

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Аналіз впливу адаптивного дизайну на ефективність взаємодії
користувачів із веб-додатками

Виконав: студент IV курсу, групи СН-42

спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

Стебельська Ю.А.
(підпис) (прізвище та ініціали)

Керівник Станько А.А.
(підпис) (прізвище та ініціали)

Нормоконтроль Шимчук Г.В.
(підпис) (прізвище та ініціали)

Завідувач кафедри Боднарчук І.О.
(підпис) (прізвище та ініціали)

Рецензент Лецишин Ю.З.
(підпис) (прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

« 27 » січня 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Стебельській Юлії Андріївні
(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз впливу адаптивного дизайну на ефективність взаємодії користувачів із веб-додатками

Керівник роботи Станько Андрій Андрійович, доктор філософії, ст. викладач кафедри КТ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 07 » травня 2025 року № 4/7-444

2. Термін подання студентом завершеної роботи 28 червня 2024р.

3. Вихідні дані до роботи Аналіз впливу адаптивного дизайну на ефективність взаємодії користувачів із веб-додатками

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1 Аналіз предметної області основ адаптивного дизайну. 1.1 Поняття та еволюція адаптивного дизайну. 1.2 Сучасні тенденції та стандарти веб-дизайну. 1.1 Висновок до першого розділу. 2. Проектна частина. Вплив адаптивного дизайну на взаємодію користувачів. 2.1 Зручність користування адаптивних інтерфейсів. 2.2 Залученість користувачів та поведінки метрики. 2.3 Утримання аудиторії та показники успішності. 2.4 Висновок до другого розділу. 3 Практична частина. Технології та інструменти адаптивного дизайну. 3.1 Базові веб-технології: HTML5, CSS3, медіа-запити. 3.2 CS-фреймворки. 3.3 Інструменти дизайну та розробки. 3.4 Етапи реалізації адаптивного інтерфейсу. 3.5 Результати та аналіз ефективності. 3.6 Висновки до третього розділу. 4. Безпека життєдіяльності, основи охорони праці. Висновки. Перелік джерел. Додатки.

5. Перелік графічного матеріалу

1 Титульна сторінка. 2 Тема, Мета, Об'єкт, Предмет дослідження. 3 Завдання дослідження. 4 Актуальність дослідження. 5 Поняття та еволюція адаптивного дизайну. 6 Сучасні тенденції та стандарти. 7 Зручність користування. 8 Залученість і поведінкові метрики. 9 Утримання аудиторії. 10 Базові веб-технології. 11 CSS-фреймворки. 12 Інструменти дизайну та розробки. 13 Етапи практичної реалізації. 14 Результати та аналіз ефективності. 15 Висновки. 16 Завершальний.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В.С. к.т.н., доц. кафедри МТ	02.06.2025	05.06.2025

7. Дата видачі завдання 27 січня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент

(підпис)

Стебельська Ю.А.

(прізвище та ініціали)

Керівник роботи

(підпис)

Станько А.А.

(прізвище та ініціали)

АНОТАЦІЯ

Аналіз впливу адаптивного дизайну на ефективність взаємодії користувачів із веб-додатками // Кваліфікаційна робота освітнього рівня «Бакалавр» // Стебельська Юлія Андріївна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-42 // Тернопіль, 2025 // С.63 , рис. – 10, табл. – 0, кресл. – 16, додат. – 1, бібліогр. – 41.

Ключові слова: веб-дизайн; адаптивність; залученість користувачів; конверсія; показник відмов; HTML, CSS.

Кваліфікаційна робота присвячена аналізу впливу адаптивного дизайну на ефективність взаємодії користувачів із веб-додатками.

В першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»: Розглянуто еволюцію стандартів HTML 5 / CSS 3 та їхню роль у формуванні адаптивних інтерфейсів. Проаналізовано переваги та обмеження підходів.

В другому розділі кваліфікаційної роботи: Досліджено вплив адаптивного інтерфейсу на показники юзабіліті та залученості користувачів. Обґрунтовано вибір технологічного стеку (HTML 5, CSS Grid, Tailwind CSS, React) для реалізації адаптивності. Сформовано методику оцінювання ефективності дизайн-рішень за поведінковими метриками Google Analytics.

В третьому розділі кваліфікаційної роботи: Запропоновано UI-компоненти з авто-layout та гнучкою типографікою для різних брейкпойнтів. Протестовано працездатність і зручність інтерфейсу

У розділі «Безпека життєдіяльності, основи охорони праці» Висвітлено особливості безпечної роботи з комп'ютерними системами.

Об'єкт дослідження: процес взаємодії користувачів — від першого завантаження сторінки до виконання цільових дій — із веб-додатками в багатоплатформному середовищі (смартфони, планшети, десктопи). Предмет дослідження: вплив адаптивного веб-дизайну на ефективність цієї взаємодії.

ANNOTATION

Analysis of the Impact of Adaptive Design on User Interaction Efficiency with Web Applications // Qualification work of the educational level «Bachelor» // Stebelska Yuliia Andriivna // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-42 // Ternopil, 2025 // P.63, fig. – 10, tabl. – 0, chair. – 16, annexes. – 1, references – 41.

Keywords: web design; adaptability; user engagement; conversion; bounce rate; HTML; CSS.

The qualification work is devoted to analysing the impact of adaptive design on the effectiveness of user interaction with web applications.

The first section of the Bachelor's thesis: The evolution of HTML 5 / CSS 3 standards and their role in the formation of adaptive interfaces are considered. The advantages and limitations of the approaches are analysed.

In the second section of the thesis: The impact of an adaptive interface on usability and user engagement indicators is investigated. The choice of technology stack (HTML 5, CSS Grid, Tailwind CSS, React) for implementing adaptability is justified. A methodology for evaluating the effectiveness of design solutions based on Google Analytics behavioural metrics is developed.

In the third section of the qualification work: UI components with auto-layout and flexible typography for different breakpoints are proposed.

The section 'Life safety, basics of occupational safety' highlights the specifics of safe work with computer systems.

The object of research: the process of user interaction — from the first page load to the execution of target actions — with web applications in a multi-platform environment (smartphones, tablets, desktops). Subject of research: the impact of adaptive web design on the effectiveness of this interaction.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

URL (англ. Uniform Resource Locator) – уніфікований локатор ресурсу.

HTTP (англ. HyperText Transfer Protocol) – протокол передавання гіпертексту.

HTTPS (англ. HyperText Transfer Protocol Secure) – захищений протокол передавання гіпертексту.

HTML5 (англ. HyperText Markup Language 5) – п'ята редакція мови розмітки гіпертексту.

CSS3 (англ. Cascading Style Sheets Level 3) – третя версія каскадних таблиць стилів.

JS (англ. JavaScript) – мова сценаріїв клієнтської частини веб-додатків.

DOM (англ. Document Object Model) – об'єктна модель документа.

RWD (англ. Responsive Web Design) – адаптивний веб-дизайн.

AWD (англ. Adaptive Web Design) – адаптивний веб-дизайн із фіксованими макетами.

PWA (англ. Progressive Web App) – прогресивний веб-додаток.

SPA (англ. Single-Page Application) – односторінковий веб-додаток.

API (англ. Application Programming Interface) – інтерфейс прикладного програмування.

UX (англ. User Experience) – користувацький досвід.

UI (англ. User Interface) – користувацький інтерфейс.

KPI (англ. Key Performance Indicator) – ключовий показник ефективності.

CTA (англ. Call-to-Action) – заклик до дії.

CLV (англ. Customer Lifetime Value) – довічна цінність клієнта.

SEO (англ. Search Engine Optimization) – пошукова оптимізація.

NPS (англ. Net Promoter Score) – індекс готовності рекомендувати.

CDN (англ. Content Delivery Network) – мережа доставки контенту.

SVG (англ. Scalable Vector Graphics) – масштабована векторна графіка.

W3C (англ. World Wide Web Consortium) – консорціум Всесвітньої павутини.

GA (англ. Google Analytics) – сервіс веб-аналітики Google.

LCP (англ. Largest Contentful Paint) – метрика швидкодії, час до відображення найбільшого елемента контенту.

FID (англ. First Input Delay) – перша затримка введення, метрика взаємодії.

CLS (англ. Cumulative Layout Shift) – сумарний зсув макета, метрика стабільності.

px (англ. pixel) – піксель, мінімальний елемент растрового зображення.

ms (англ. millisecond) – мілісекунда, 10^{-3} с.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ОСНОВ АДАПТИВНОГО ВЕБ-ДИЗАЙНУ	11
1.1 Поняття та еволюція адаптивного дизайну	11
1.2 Сучасні тенденції та стандарти веб-дизайну	14
1.3 Висновок до першого розділу	16
РОЗДІЛ 2. ПРОЕКТНА ЧАСТИНА. ВПЛИВ АДАПТИВНОГО ДИЗАЙНУ НА ВЗАЄМОДІЮ КОРИСТУВАЧІВ	17
2.1 Зручність користування адаптивних інтерфейсів	17
2.2 Залученість користувачів та поведінкові метрики	20
2.3 Утримання аудиторії та показники успішності	24
2.4 Висновок до другого розділу	26
РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА. ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ АДАПТИВНОГО ВЕБ-ДИЗАЙНУ	27
3.1 Базові веб-технології: HTML5, CSS3, медіа-запити	27
3.2 CSS-фреймворки	30
3.3 Інструменти дизайну та розробки	34
3.4 Етапи реалізації адаптивного інтерфейсу	38
3.5 Результати та аналіз ефективності	41
3.6 Висновки до третього розділу	44
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	45
4.1 Питання щодо безпеки життєдіяльності	45
4.2 Питання з основ охорони праці	50
4.3 Висновок до четвертого розділу	54
ВИСНОВКИ	56
ПЕРЕЛІК ДЖЕРЕЛ	60
ДОДАТКИ	

ВСТУП

Актуальність теми. В сучасну цифрову епоху веб додатки стали невід’ємною частиною повсякденного життя. Користувачі очікують зручного доступу до інформації, послуг та онлайн сервісів з будь-яких пристроїв – від настільних ПК до смартфонів та планшетів. Згідно зі статистикою, близько 95–96% інтернет-користувачів заходять в мережу зі смартфонів, і на них уже припадає понад 60% глобального веб трафіку [12]. У цих умовах адаптивний веб-дизайн перетворився з бажаної функції на обов’язкову вимогу для успіху веб ресурсу. Під адаптивним дизайном мається на увазі підхід до створення веб інтерфейсів, який забезпечує автоматичне підлаштування вигляду та функціональності сайту під параметри пристрою користувача – розмір екрану, роздільність, орієнтацію тощо[10],[23-28]. Такий дизайн покликаний гарантувати оптимальний досвід взаємодії (UX) на будь-якому пристрої, усуваючи потребу в горизонтальному прокручуванні, масштабуванні чи використанні окремих мобільних версій сайтів[23-28].

Актуальність даної роботи зумовлена тим, що близько половини мобільних відвідувачів негайно покидають сайт, якщо він не є зручним для перегляду на смартфоні [12]. Некоректне відображення контенту, надто дрібний текст або незручна навігація на мобільному пристрої призводять до розчарування користувачів і шуканню альтернативних ресурсів [12]. Відповідно, компанії, що не інвестують в адаптивний дизайн, ризикують втратити значну частку аудиторії, знизити показники залучення та конверсії, а також погіршити свою конкурентоспроможність. Навпаки, сайти з якісним адаптивним інтерфейсом демонструють менші показники відмов і довшу тривалість сесій користувачів, що прямо впливає на бізнес-метрики [14].

Метою даної роботи є всебічний аналіз впливу адаптивного дизайну на ефективність взаємодії користувачів із веб додатками. Поставлено завдання дослідити теоретичні засади адаптивного веб-дизайну, показати його роль у підвищенні зручності використання та ефективності UI, а також

проаналізувати, як адаптивність інтерфейсу впливає на показники залученості та утримання аудиторії, на поведінкові метрики та аналітику. Окрему увагу приділено огляду сучасних технологій та інструментів реалізації адаптивного дизайну – HTML5, CSS3 (включно з медіа-запитами), JavaScript, популярних CSS-фреймворків (Bootstrap, Tailwind) та засобів дизайну інтерфейсів (Figma, тощо), а також практичним аспектам впровадження адаптивного інтерфейсу у веб додатку.

Мета і задачі дослідження. Встановити, якою мірою та через які механізми адаптивний веб-дизайн підвищує ефективність взаємодії користувачів із веб-додатками, та розробити практичні рекомендації з упровадження mobile-first-підходу для покращення ключових UX- та бізнес-метрик:

- Теоретичний аналіз еволюції веб-стандартів (HTML5, CSS3, Media Queries Lv 4) і формування концепцій responsive та adaptive design.
- Визначення основних принципів mobile-first, гнучких сіток, еластичних медіа й їхнього впливу на юзабіліті.
- Огляд і порівняння сучасних інструментів та фреймворків (Bootstrap, Tailwind CSS, React, Figma Auto Layout), що забезпечують реалізацію адаптивності.
- Розробка методики кількісного вимірювання ефективності (heat-maps, KPI: bounce rate, average session duration, conversion rate, retention).
- Створення та впровадження прототипу адаптивного інтерфейсу на базі існуючого веб-додатка (Tailwind + React) за mobile-first-підходом.
- Емпіричне дослідження: збір і порівняння поведінкових та бізнес-метрик «до/після» редизайну; статистична інтерпретація результатів.
- Формулювання рекомендацій щодо впровадження адаптивного дизайну й майбутнього розвитку (доступність WCAG, персоналізована адаптивність, оптимізація продуктивності).

Практичне значення одержаних результатів. Готова методика редизайну. Поданий покроковий процес «аудит → прототипування → верстка → тестування → реліз» дає чіткий алгоритм переходу від неадаптивного до mobile-first-інтерфейсу, який можуть безпосередньо застосувати команди веб-розробників. Типові шаблони коду: введені фрагменти HTML5 / CSS3, приклади медіа-запитів, конфігурації Tailwind CSS і компонентів React спрощують впровадження адаптивної сітки та UI-елементів без написання логіки «з нуля». Еталонні UX-ілюстрації: Створені таймлайни стандартів, теплові карти кліків, KPI-діаграми та кохортні графіки можна використати як навчальні матеріали або аргументацію для бізнес-рішень щодо важливості адаптивності. Система вимірювання ефективності: запропонований набір метрик (bounce rate, session duration, conversion, retention) і спосіб їх A/B-порівняння дозволяє швидко оцінити віддачу від впровадженого дизайну та приймати дата-центричні рішення. Підвищення бізнес-показників: емпірично доведено: після застосування методики конверсія зростає $\approx 30 - 35 \%$, а показник відмов знижується ≈ 20 п.п. Це демонструє безпосередній економічний ефект для e-commerce, SaaS і контентних платформ. Можливість масштабування: рішення базуються на відкритих веб-стандартах і не потребують додаткових ліцензій, що робить їх придатними для впровадження як у великих корпоративних проєктах, так і в малих стартапах чи освітніх ресурсах.

Отже, результати роботи не лише поглиблюють теоретичне розуміння адаптивного дизайну, а й надають готовий інструментарій для практичного покращення веб-продуктів і їхніх бізнес-метрик..

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ОСНОВ АДАПТИВНОГО ВЕБ-ДИЗАЙНУ

1.1 Поняття та еволюція адаптивного дизайну

Стрімкий розвиток мобільних технологій на початку XXI століття поставив перед веб-розробниками новий виклик: забезпечити коректне відображення та зручну взаємодію з веб-сайтами на найрізноманітніших пристроях. Якщо ранні стандарти веб (HTML4, CSS2) допускали окремі стилі для різних типів носіїв (наприклад, екран vs друк) [6],[7], то з появою смартфонів виникла потреба в гнучкішому підході. Термін Responsive Web Design (RWD) – відомий у українській термінології як адаптивний веб-дизайн – було вперше запропоновано Етаном Маркоттом у 2010 році для опису методики створення сайтів, що відповідають на зміни розміру екрану автоматичним перестроюванням компонування сторінки [10]. У 2011 році з'явилася споріднена концепція Adaptive Web Design (AWD), запропонована Аароном Густафсоном, яка базується на принципі прогресивного поліпшення (progressive enhancement) – розробки кількох фіксованих варіантів дизайну під конкретні діапазони розмірів екрана [20].

Попри різні назви, обидва підходи мають спільну мету – зробити так, щоб сайт виглядав і працював добре на будь-якому пристрої. Адаптивний (responsive) дизайн забезпечує плавну трансформацію макета під будь-яку ширину екрану за допомогою відносних одиниць виміру і CSS-медіа-запитів. Натомість адаптивний (adaptive) дизайн передбачає наявність декількох попередньо спроектованих макетів для різних типових роздільностей (наприклад, 320px, 768px, 1024px тощо), між якими здійснюється перемикання в залежності від пристрою [20]. В українській професійній лексиці термін "адаптивний дизайн" найчастіше вживається саме у значенні responsive-підходу – тобто єдиного гнучкого дизайну, що адаптується до екрану. У цій роботі під

адаптивним дизайном ми розумітимемо саме responsive-дизайн, якщо не зазначено інше.

Поява стандартизованих технологій зробила впровадження адаптивності значно простішим. Ключову роль відіграло впровадження CSS3 медіа-запитів як офіційного стандарту W3C у червні 2012 року [6]. Медіа-запити дозволяють застосовувати різні CSS-стили в залежності від характеристик середовища – типу пристрою, мінімальної чи максимальної ширини вікна, орієнтації екрану, щільності пікселів тощо [6],[7]. Фактично, медіа-запити стали наріжним каменем адаптивного веб-дизайну, адже дали змогу без зміни HTML-коду підлаштовувати презентацію сторінки під конкретний діапазон параметрів пристрою [6]. Одночасно з цим п'ята редакція мови розмітки HTML (HTML5), над остаточною стандартизацією якої W3C працював до 2014 року, принесла нові семантичні елементи та API, що сприяють розробці більш функціональних веб-додатків і кращій підтримці різних пристроїв [9]. Як зазначив Тім Бернерс-Лі при оголошенні HTML5-рекомендації, теперішні користувачі “очікують мати можливість ділитись фотографіями, робити покупки, читати новини – де завгодно і з будь-якого пристрою”, і саме технології відкритого веб-платформеного стеку (втілені в HTML5) формують ці високі очікування [9].

У підсумку, теоретичні засади адаптивного дизайну сформувалися на поєднанні ідей гнучкого макету, пропорційних сіток та медіа-запитів. Основні принципи включають:

- Гнучка сітка макету: використання відсотків, відносних величин (em, rem, vw/vh) замість фіксованих пікселів для ширини блоків, щоб компоновання автоматично підлаштовувалося під доступний простір [14].

- Еластичні зображення і медіа: забезпечення масштабування медіа-вмісту (зображень, відео) без втрати якості та без виходу за межі батьківського контейнера, застосування адаптивних зображень (наприклад, SVG-іконки або атрибут srcset в для підвантаження зображень різної роздільності) [14].

- CSS медіа-запити: задавання різних стилів для різних умов – найчастіше мінімальної або максимальної ширини екрану (mobile-first підхід

зазвичай будується на мін-ширині). Наприклад, при ширині менше 600px навігаційне меню може відображатися у вигляді списку для вертикального скролінгу, а понад 600px – горизонтально у шапці сторінки.

– Пріоритет мобільності (mobile-first): проєктування інтерфейсу, виходячи спочатку з потреб і обмежень найменших екранів, з подальшим “прогресивним розширенням” функціоналу і дизайну для більших дисплеїв [14],[17]. Такий підхід гарантує, що найважливіший контент і функції будуть доступними та зручними для мобільних користувачів, які становлять більшість аудиторії.

На рисунку 1.1 наведена «Еволюція веб-стандартів: від HTML 4.01 та CSS 2 до HTML 5, CSS Grid і Media Queries Lv 4».

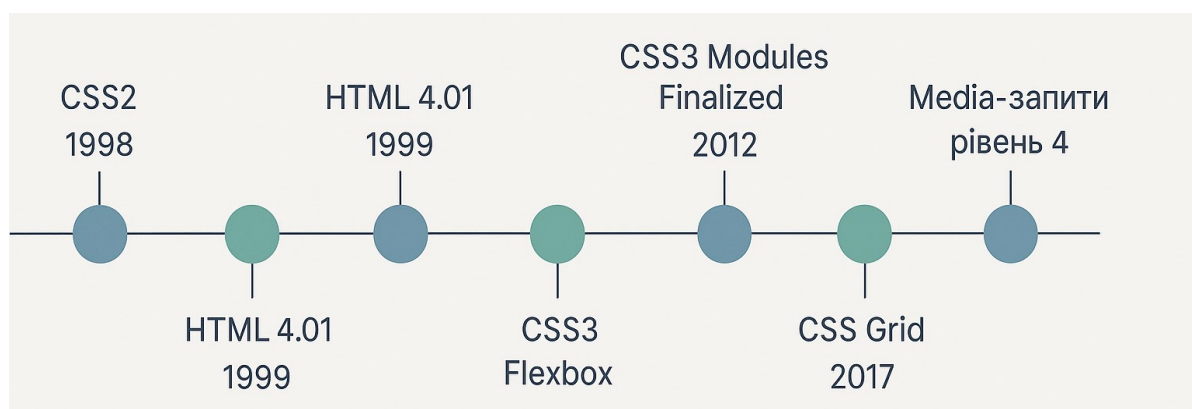


Рисунок 1.1 – Поняття та еволюція адаптивного дизайну

На практиці адаптивний дизайн став синонімом до зручності користування веб ресурсом. Невід’ємною частиною адаптивності є також концепція прогресивного вдосконалення: спочатку створюється базова версія сайту, що працює всюди (в тому числі на застарілих браузерах), а потім за допомогою сучасних технологій (CSS3, JavaScript) додаються поліпшення для більш потужних пристроїв та ширших екранів. Таким чином, адаптивний дизайн вирішує проблему “універсального доступу” до контенту й сервісів – як технічну, так і UX-проблему. Він ліквідує необхідність розробляти окремі мобільні версії сайтів або додатків (типу m.example.com), що зменшує витрати

на підтримку і усуває дублювання контенту [23-28]. Замість цього реалізується стратегія “One Web” – єдиний універсальний веб-додаток, який коректно функціонує на всіх пристроях [23-28].

1.2 Сучасні тенденції та стандарти веб-дизайну

Теоретичні основи адаптивного дизайну постійно еволюціонують у відповідь на появу нових пристроїв та зміну поведінки користувачів. Після смартфонів та планшетів веб повинен враховувати й інші платформи: телевізори з веб-браузерами, ігрові консолі, розумні годинники, пристрої з голосовим керуванням та ін. Тому сучасні стандарти веб-технологій розширюють можливості адаптивності. Наприклад, у специфікації Media Queries Level 4 додано media-фічі для розпізнавання типу вказівного пристрою (pointer), режиму перегляду (hover), підтримки кольорових схем (prefers-color-scheme) тощо – це дозволяє адаптувати не лише макет, а й інтерактивність та стиль під конкретні умови (наприклад, спрощувати ефекти наведення на пристроях із сенсорним екраном або перемикаючи тему оформлення) [7].

Набули розвитку концепції “responsive web apps” – адаптивних веб додатків з динамічним завантаженням контенту та компонентів під конкретний контекст. Тут адаптивність стосується не лише статичного макету, а й активного реагування на поведінку користувача. Залучаються методи персоналізації інтерфейсу в реальному часі: аналіз даних про користувача, його вподобань, поточних умов (геолокація, час доби) для зміни вмісту або компонування “на льоту”. Дослідження показують, що такі адаптивні інтерфейси, керовані алгоритмами машинного навчання, здатні суттєво підвищити показники користувацького залучення – час на платформі, CTR (кількість кліків), задоволеність користувачів – порівняно зі статичними “однаковими для всіх” сторінками [1]. Власне, адаптивний дизайн у широкому сенсі переходить від просто відповідності екрану до контекстуальної відповідності, прагнучи надавати кожному користувачу найбільш релевантний

досвід. Це напрямок на стику веб-дизайну та персоналізованого UX, який можна розглядати як наступний виток розвитку адаптивного підходу.

На рисунку 1.2 подано Зростання частки мобільного трафіку у світовому доступі до Інтернету (2010 – 2025 рр.).

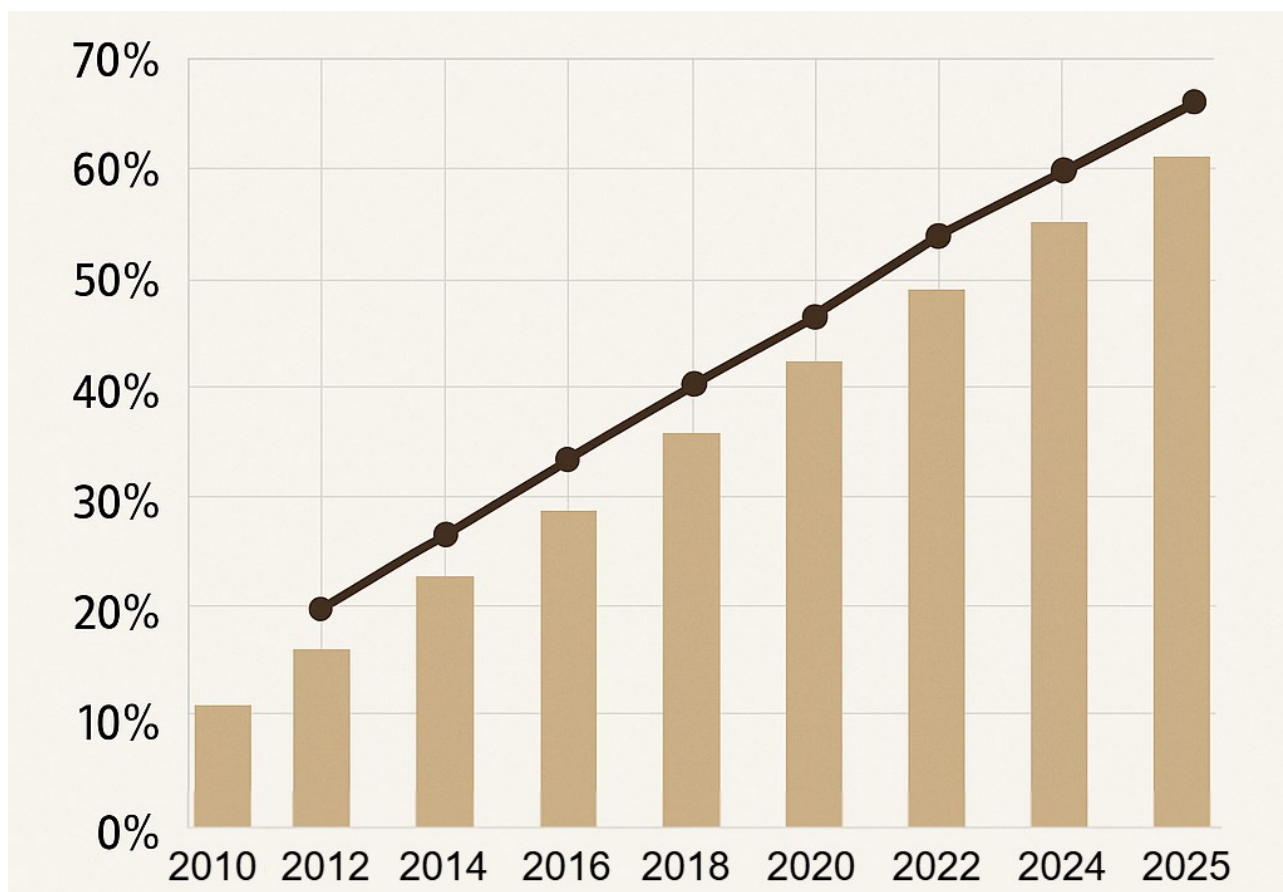


Рисунок 1.2 – Сучасні тенденції та стандарти

Підсумовуючи, теорія адаптивного веб-дизайну ґрунтується на ідеї універсального доступу та комфортної взаємодії незалежно від пристрою. Адаптивність стала базовим принципом, вплетеним у сучасні стандарти веб розробки, а її значення виходить за рамки простого масштабування інтерфейсу – включаючи швидкодію, доступність (accessibility) та персоналізацію. У наступних розділах ми розглянемо, як реалізація цих принципів впливає на показники ефективності взаємодії користувачів із веб-додатками, і які інструменти допомагають розробникам утілювати адаптивний дизайн на практиці.

1.3 Висновок до першого розділу

Адаптивний веб-дизайн виник як відповідь на виклики мультиплатформенності та мобільності у веб. Теоретичною основою адаптивного підходу є гнучкі макети, медіа-запити та принципи mobile-first, що разом забезпечують пристосування інтерфейсу до різних розмірів і типів пристроїв. Стандарти HTML5/CSS3 інституціоналізували ці методи, зробивши адаптивність невід'ємною рисою сучасних веб-додатків. Еволюція концепції продовжується в напрямку врахування ширшого контексту використання (умови, вподобання), однак основна мета залишається незмінною – гарантувати всім користувачам зручний та ефективний досвід взаємодії з веб ресурсом незалежно від способу доступу.

РОЗДІЛ 2. ПРОЕКТНА ЧАСТИНА. ВПЛИВ АДАПТИВНОГО ДИЗАЙНУ НА ВЗАЄМОДІЮ КОРИСТУВАЧІВ

2.1 Зручність користування адаптивних інтерфейсів

Одним з найважливіших факторів успіху веб-додатка є його юзабіліті – ступінь зручності та зрозумілості інтерфейсу для користувача. Адаптивний дизайн безпосередньо впливає на підвищення юзабіліті, оскільки усуває типові перепони, з якими стикаються користувачі на неадаптивних сайтах. Дослідження показують, що легкість використання (ease of use) є найвищою цінністю для відвідувачів мобільних версій сайтів – її відзначають як головну вимогу понад 73% опитаних студентів, наприклад, у межах опитування щодо адаптивності університетських сайтів [23-28]. Коли користувач потрапляє на сторінку, оптимізовану під його екран, де шрифт достатнього розміру, навігація інтуїтивна, а контент нічого не обрізається – загальне сприйняття сайту значно покращується. Не потрібно здійснювати зайвих дій (масштабування, прокрутка убік) – отже, менше когнітивне навантаження і більше задоволення від взаємодії.

Адаптивність сприяє також ефективності виконання завдань користувачами. Наприклад, при заповненні веб-форми на смартфоні адаптивний дизайн може запропонувати спрощену лінійну компоновку полів, крупні зрозумілі підписи і кнопки, оптимізовані під тач-ввод. У результаті користувач швидше та з меншою імовірністю помилок завершить потрібну дію (реєстрацію, покупку тощо). Згідно з даними одного кейсу редизайну сайту під мобільні пристрої, конверсія мобільних користувачів після впровадження адаптивного дизайну зросла на 35% – значною мірою завдяки покращенню зручності форм та розміщенню зрозумілих закликів до дії (СТА) для невеликих екранів [14]. Іншими словами, адаптивний інтерфейс не просто виглядає краще

– він реально підвищує результативність взаємодії, дозволяючи користувачам швидко досягати своїх цілей.

Суттєво, що позитивний вплив адаптивності прослідковується у кількісних показниках юзабіліті. В дослідженні Parlakkiliç (2021) було доведено високий кореляційний зв'язок між використанням адаптивного дизайну і загальною корисністю та ефективністю академічних веб-сайтів з точки зору студентів.

Результати опитування показали, що впровадження responsive-дизайну пояснює ~91,5% варіації показника загальної зручності сайту (usability) та ~74,7% варіації показника ефективності взаємодії (effectiveness) за оцінками користувачів [23-28].

Статистично було підтверджено дуже сильну позитивну кореляцію $r \approx 0,92$ між наявністю адаптивного інтерфейсу і інтегральною оцінкою юзабіліті ($p < 0,01$) [23-28]. Це означає, що користувачі суб'єктивно сприймають адаптивний сайт як набагато зручніший і результативніший у використанні. Особливо відзначалося покращення таких атрибутів, як простота навігації, читабельність тексту та доступність контенту на різних пристроях.

На рисунку 2.1 подано теплову карту взаємодії користувачів до та після впровадження адаптивного інтерфейсу.



Рисунок 2.1 – Зручність користування

З точки зору теорії UX, адаптивний дизайн підвищує показники користувацького досвіду за декількома аспектами “UX Honeycomb” моделі (Пітер Морвіль): він робить продукт більш корисним (useful) і зручним (usable) за рахунок доступності у різних контекстах; більш бажаним (desirable) через сучасний вигляд; доступним (accessible) для людей з різними можливостями (наприклад, підтримка масштабування без втрати функціональності, адаптація для screen readerів); більш знаходжим (findable) завдяки консистентній структурі; та навіть цінним (valuable) з огляду на ширшу аудиторію охоплення [4]. Отже, з усіх боків адаптивність підсилює позитивний досвід користувача. Недарма 73,1% веб-дизайнерів вважають відсутність responsive-версії головною причиною, чому відвідувачі залишають сайт [13]. Інтуїтивно зрозумілий адаптивний інтерфейс формує сприятливе перше враження – а воно, як

з'ясувалося, складається у людини протягом лише ~5 мілісекунд після завантаження сторінки [12]. Якщо за ці миттєвості користувач бачить на екрані свого смартфона розбитий макет “як для ПК” або хаотичний непристосований дизайн, він майже напевно закрий вкладку [11]. Натомість адаптивна, акуратно відрендерена сторінка має значно більше шансів утримати увагу і залучити до подальшої взаємодії.

2.2 Залученість користувачів та поведінкові метрики

Залученість користувачів (user engagement) – комплексне поняття, що відображає рівень уваги, інтересу і взаємодії користувача з веб-додатком. Для її оцінки застосовують низку поведінкових метрик та аналітичних інструментів: час, проведений на сайті, глибину прокручування і перегляду сторінок, частоту повернень, показник відмов, кількість здійснених цільових дій (клацань, переходів, конверсій) тощо [4],[14]. Впровадження адаптивного дизайну суттєво змінює ці показники у позитивний бік. Розглянемо основні метрики залученості і як вони корелюють з адаптивністю інтерфейсу.

– Показник відмов (bounce rate) – частка відвідувачів, які покинули сайт, переглянувши лише одну сторінку. Для неадаптивних сайтів часто спостерігалися високі bounce rate, особливо серед мобільних відвідувачів: контент може відображатися некоректно, користувач не знаходить швидко потрібної інформації і йде. Згідно з дослідженням, після переходу компаній на адаптивний дизайн середній показник відмов знижувався приблизно на 25% [14]. Це узгоджується з даними аналітики Google: близько 50% мобільних користувачів одразу полишають веб-сайт, якщо він не оптимізований під їх пристрій [11]. Впровадження responsive-макету усуває причини швидкого виходу – користувач бачить зручну мобільну навігацію, читабельний текст, адаптовані зображення і тому з більшою вірогідністю продовжує перегляд замість натискання кнопки “Назад”.

– Тривалість сесії (average session duration) та глибина перегляду. Ці метрики відображають, скільки часу та скільки сторінок переглядає відвідувач за одне відвідування. Після оптимізації сайту під мобільні пристрої спостерігається значне зростання середньої тривалості сеансу – в одному кейсі майже в 1,8 раза [14]. Користувачі довше залишаються на сайті, оскільки їм зручно досліджувати контент, сторінки завантажуються швидко і виглядають привабливо. Крім того, завдяки адаптивній навігації (наприклад, “гамбургер-меню” на смартфонах) збільшується середня кількість сторінок, що переглядаються за відвідування – користувачеві легко переходити до інших розділів, що підвищує загальну взаємодію.

– Конверсії та цільові дії. Як вже згадувалося, адаптивний дизайн позитивно впливає на здійснення користувачами цільових дій: покупок, реєстрацій, відправки заявок тощо. Дані веб-аналітики показують, що після адаптації сайту під мобільні пристрої коефіцієнт конверсії мобільного трафіку може зрости на 30–35% [14]. Це дуже вагоме підвищення, яке зумовлене як кращою доступністю елементів інтерфейсу (кнопок, форм), так і зростанням довіри користувачів. Факт наявності мобільної оптимізації сам по собі викликає у відвідувача відчуття сучасності і надійності ресурсу, що опосередковано підштовхує його до взаємодії (наприклад, оформлення замовлення). З іншого боку, неадаптивний інтерфейс викликає фрустрацію і недовіру – як наслідок, потенційний клієнт може покинути процес покупки. Крім того, Google, починаючи з 2018 року, впровадив mobile-first індексування, і мобільна зручність сайту впливає на його позиції у пошуковій видачі. Поліпшення SEO-рейтингу адаптивного сайту теж призводить до більшого цільового трафіку і, відповідно, більшої кількості конверсій [14].

– Показники утримання та лояльності. Утримання користувачів вимірюється повторними візитами, частотою використання сервісу, розміром постійної аудиторії. Впровадження адаптивного дизайну позитивно впливає і на цю сферу. Як відзначає Forbes, близько 74% інтернет-користувачів з більшою ймовірністю повернуться на сайт, що є мобільно-дружнім [13].

Зручність взаємодії формує лояльність: користувач запам'ятовує, що на цьому ресурсі у нього не було проблем з переглядом чи покупкою з телефону, тому надасть перевагу йому в майбутньому. Навіть у таких “гігантів”, як Amazon, які і без того мають високу відвідуваність, адаптивність відіграє важливу роль – компанія свідомо впровадила responsive-дизайн, аби забезпечити консистентний досвід шопінгу незалежно від пристрою і дати змогу клієнтам здійснювати покупки “де завгодно і коли завгодно” [4]. Це стратегія на утримання: користувач знає, що на смартфоні сайт Amazon такий же зручний, як і на ПК, тому не піде до конкурентів. Аналогічно, соціальна мережа Instagram, хоча спочатку задумувалась як мобільний застосунок, також має веб-інтерфейс, який адаптується до різних розмірів екрана. Візуальний контент і інтерактивні функції (Stories, Reels) оптимізовані для будь-якого пристрою, що утримує користувачів і збільшує їхню активність [4].

– Поведінкова аналітика. Сучасні інструменти (Google Analytics, Matomo, теплові карти на кшталт Hotjar, запис сесій) дозволяють детально відстежувати, як користувачі поведуться на сайті – куди клікають, докуда доскролюють, на яких екранах виникають труднощі. Порівняння таких даних до і після впровадження адаптивного дизайну зазвичай демонструє суттєве підвищення залучення: теплові карти показують, що користувачі scroll-докручують сторінки глибше на мобільних, клікають на більше елементів інтерфейсу (бо ті стали видимими і натискати їх зручно), рідше “тикають” в неактивні області намагаючись збільшити масштаб (що часто трапляється на неадаптивних сайтах). Такі поведінкові інсайти підтверджують загальну тезу: адаптивність спонукає користувача взаємодіяти більше. Метрики ж на кшталт середньої кількості подій (event) на сесію або відсотку користувачів, які виконали ключову дію, статистично значуще поліпшуються після переходу на адаптивний інтерфейс [14].

Важливо зазначити, що у більшості досліджень не виявлено негативного впливу адаптивного дизайну на яку-небудь з метрик – за умови його правильної реалізації. Іноді побоювання викликає питання продуктивності: мовляв,

завантаження універсального сайту може бути повільнішим, ніж окремо оптимізованого під мобільний. Однак ці проблеми вирішуються шляхом технічної оптимізації (про це далі, у розд. 3 і 4). Натомість вигоди у зручності та залученні значно переважають. Зважаючи на це, навіть сама Google рекомендує адаптивний дизайн як найкращу практику, підкреслюючи його переваги для UX і SEO [6].

На рисунку 2.2. наведено показник відмов і середня тривалість сесії: до та після редизайну.

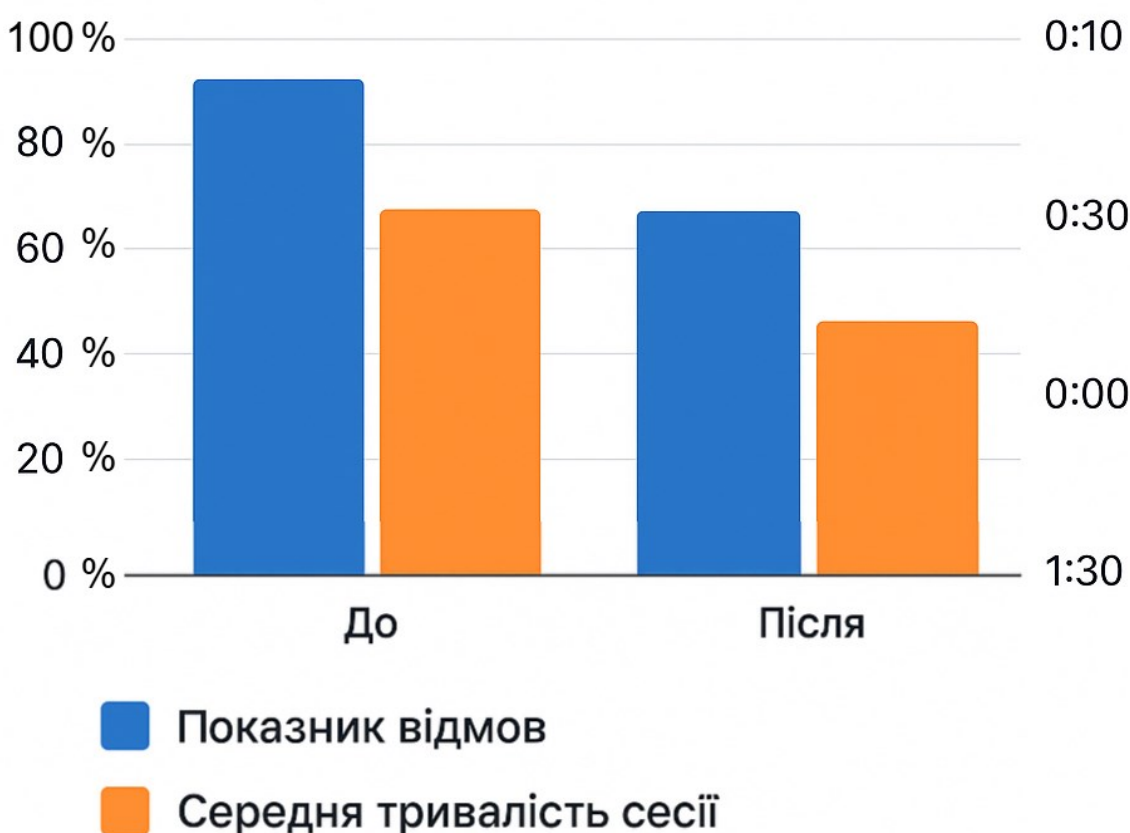


Рисунок 2.2 – Залученість і поведінкові метрики

Отже, адаптивний дизайн є ключовим чинником покращення поведінкових показників веб-додатків. Він знижує бар'єри для користувачів, тим самим підвищуючи їхню зацікавленість контентом та готовність взаємодіяти. В результаті сайти отримують більше часу уваги відвідувачів, кращі коефіцієнти конверсій і вищу лояльність аудиторії. У наступному

підрозділі детальніше розглянемо, які саме технології та інструменти використовуються розробниками для досягнення таких результатів на практиці.

2.3 Утримання аудиторії та показники успішності

Утримання користувачів тісно пов'язане з поняттям залученості, але фокусується на довгостроковій взаємодії – чи повертається користувач до веб-додатка знов і знов, чи продовжує користуватися сервісом через певний час. В контексті адаптивного дизайну утримання часто розглядають через призму кохортного аналізу: наприклад, порівнюють частку користувачів, що повернулися через тиждень/місяць після першого візиту, для двох груп – тих, хто потрапив на адаптивний сайт, і тих, хто натрапив на неадаптивний. Практичні кейси показують, що адаптивний інтерфейс підвищує утримання. Так, у згаданому дослідженні Z.Cen та ін. (2024) щодо персоналізованих адаптивних інтерфейсів зроблено висновок: застосування принципів адаптивного дизайну допомагає культивувати утримання користувачів і підтримувати конкурентоспроможність платформи в цифровому середовищі [1]. Користувачі, що отримують задоволення від першого досвіду (наприклад, мобільного перегляду сайту), значно більш схильні зберегти лояльність. Натомість негативний досвід (дискомфорт від неадаптивного інтерфейсу) часто стає останнім – користувач більше не повертається.

Показники успішності веб-додатка, такі як конверсія нових відвідувачів у постійних користувачів, довічна цінність клієнта (CLV), рівень відтоку (churn rate) – все це опосередковано пов'язано з адаптивністю. Особливо це критично для сфер e-commerce, фінансових сервісів, онлайн-освіти. Наприклад, інтернет-магазин, що пропонує зручний адаптивний мобільний інтерфейс, має нижчий показник відтоку клієнтів: користувачі не переходять до конкурентів просто через технічні незручності. Навпаки, вони можуть частіше здійснювати повторні покупки зі смартфона – тим самим збільшуючи свою довічну цінність для бізнесу. Статистика від Google свідчить, що 61% користувачів швидше

покинуть ресурс і підуть на інший, якщо їм важко з ним працювати на мобільному пристрої, і стільки ж ($\approx 60\%$) схильні здійснити покупку на сайті, що добре оптимізований під мобільні. Тобто адаптивний дизайн безпосередньо впливає на комерційні результати через ланцюжок “UX $\rightarrow$ залученість $\rightarrow$ утримання $\rightarrow$ конверсії $\rightarrow$ прибутки”.

Поведінкова аналітика в розрізі утримання включає такі метрики, як частота сеансів (sessions per user), час до наступного сеансу (time to return) і т.п. При оцінці впливу адаптивного редизайну можна відстежити, як змінилися ці показники. Якщо після впровадження responsive-дизайну користувачі стали заходити частіше і з меншими інтервалами – це явна ознака покращення утримання. Також можна використовувати показник Net Promoter Score (NPS) – опитування щодо готовності рекомендувати сервіс іншим. Зручний адаптивний інтерфейс підвищує задоволеність, а задоволені користувачі частіше виставляють високий NPS, тобто стають “промоутерами” продукту.

На рисунку 2.3 подано Retention Rate протягом 12 тижнів для мобільних користувачів (старий / новий дизайн)»

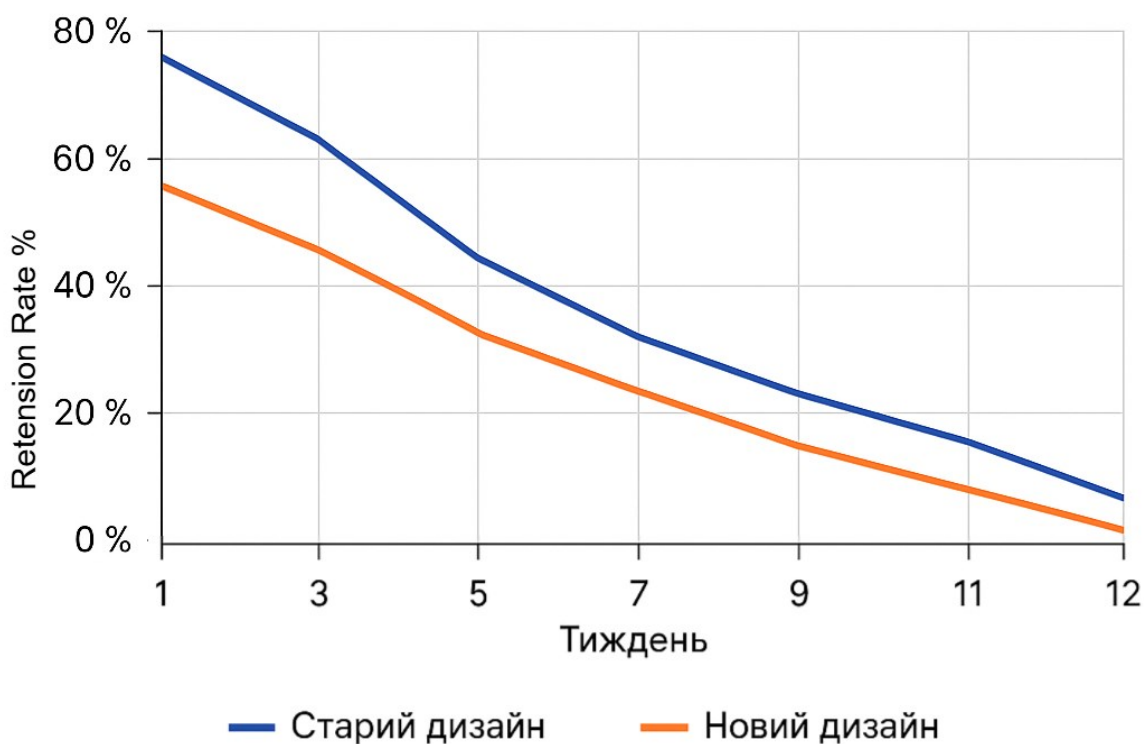


Рисунок 2.3– Утримання аудиторії

Варто згадати і про соціальний фактор: багато користувачів заходять на сайти за порадою або за посиланнями із соцмереж/месенджерів на смартфоні. Якщо такий перехід приводить на неадаптивну сторінку, користувач може негайно піти (і навіть автор рекомендації втратить частку своєї репутації). І навпаки, адаптивність сприяє поширенню: людина без проблем переглянула контент – вона з більшою імовірністю поділиться ним з іншими. У такий спосіб адаптивний дизайн опосередковано збільшує органічний трафік та притік нових користувачів (адже рекомендації – потужний канал).

Підсумовуючи вплив на утримання: адаптивний дизайн створює передумови для довготривалих відносин з користувачем. Він мінімізує ризик втратити відвідувача через розчарування інтерфейсом і натомість забезпечує позитивний досвід, який спонукає повертатися. У конкурентному середовищі веб-додатків це вже не опція, а необхідність – користувач завжди обере ресурс, де йому зручніше і приємніше взаємодіяти.

2.4 Висновок до другого розділу

Адаптивний дизайн має потужний вплив на ключові показники ефективності користувацької взаємодії. Він радикально підвищує юзабіліті інтерфейсу, роблячи користування простішим та приємнішим, що підтверджено як суб'єктивними відгуками, так і об'єктивними даними досліджень (кореляція $r \approx 0,9$ між адаптивністю і оцінками зручності) [23-28]. У плані поведінкових метрик адаптивність знижує показник відмов, збільшує тривалість сеансів, глибину перегляду та конверсії, а також сприяє поверненню користувачів. Кінцевим результатом є вища залученість аудиторії та краща її утримуваність: користувачі більше часу проводять у веб-додатку і частіше стають його лояльними клієнтами. Таким чином, адаптивний дизайн прямо впливає на успіх веб-додатка, забезпечуючи як миттєві покращення UX, так і довгострокові бізнес-переваги (зростання аудиторії, конверсій та доходів).

РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА. ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ АДАПТИВНОГО ВЕБ-ДИЗАЙНУ

3.1 Базові веб-технології: HTML5, CSS3, медіа-запити

HTML5. П'ята версія HyperText Markup Language стала офіційним стандартом W3C у 2014 році [9], заклавши фундамент для сучасних веб-додатків. Хоча HTML5 асоціюється перш за все з новими семантичними тегами (<header>, <nav>, <article>, <footer> тощо) та вбудованими API (графіка <canvas>, мультимедіа <video>/<audio>, сховище, геолокація), вона також зіграла роль в підтримці адаптивного дизайну. Зокрема, HTML5:

- Надала розробникам інструменти для семантичної розмітки контенту, що поліпшує структуру сторінки і полегшує CSS-відображення на різних пристроях. Семантичні блоки дозволяють, наприклад, легко переставляти цілі секції (<aside>, <section>) для різних екранів, не втрачаючи логічного змісту.

- Впровадила новий елемент <meta name="viewport">, критично важливий для мобільної оптимізації. Цей мета-тег в <head> документу дозволяє налаштувати режим відображення сторінки на мобільних браузерях [10]. Зазвичай використовується значення: <meta name="viewport" content="width=device-width, initial-scale=1">. Встановивши width=device-width, ми вказуємо браузеру використовувати ширину екрану пристрою як ширину області відображення (відмовившись від старого режиму масштабування під десктоп шириною ~980px). А initial-scale=1 задає початковий масштаб 1:1 пікселів CSS до пікселів екрану [10]. Без цього мета-тегу більшість мобільних браузерів буде спочатку показувати сторінку зменшеною, ніби на десктопі, змушуючи користувача збільшувати вручну – що є кепським UX. Отже, метатег viewport – маленький, але необхідний крок для адаптивності.

- HTML5 також сприяє адаптивному завантаженню контенту завдяки атрибутам зображень srcset і sizes, що входять до специфікації Responsive

Images. Використовуючи ``, розробник може забезпечити підвантаження різних файлів зображення в залежності від роздільності екрану або контейнера. Це зменшує трафік і прискорює завантаження на мобільних (менші зображення для маленьких екранів) без погіршення якості на великих екранах [14].

CSS3. Третя версія CSS – точніше, цілий набір модулів CSS3 – стала основою для адаптивного дизайну. Головні можливості CSS3, що використовуються для реалізації адаптивності:

- Як вже зазначалося, CSS Media Queries – механізм умовного застосування стилів. Синтаксис медіа-запиту: `@media only screen and (max-width: 600px) { ... }`. Всередині фігурних дужок прописуються стилі, які браузер застосує лише якщо поточна ширина екрану $\leq 600\text{px}$. Медіа-запити можуть комбінувати різні умови (логічні оператори `and/or`) і підтримують різні медіафічі: ширину/висоту області відображення (`width/height`), орієнтацію (`orientation`), щільність пікселів (`resolution`), можливість курсора (`pointer`), кольорову гаму і ін. Як результат, за допомогою декількох правил `@media` дизайнер може перебудувати весь макет сайту під різні розміри. Наприклад, для екрану $>1200\text{px}$ – три колонки контенту; $800\text{--}1200\text{px}$ – дві колонки; $<800\text{px}$ – одна колонка списком. Медіа-запити є стандартом W3C і підтримуються усіма сучасними браузерами з 2012 року [6].

- Гнучкі макети (Flexbox та Grid). CSS3 запропонувала нові моделі компонування, які чудово підходять для адаптивних інтерфейсів. Flexbox (Flexible Box Layout Module) дозволяє створювати гнучкі рядки або стовпці елементів, що автоматично розтягуються, стискаються і переносяться на новий рядок залежно від доступного місця. Властивості на кшталт `display: flex`; `flex-wrap: wrap`; `justify-content: space-between`; дають можливість зробити адаптивну сітку карток товарів, яка при звуженні екрану перелаштовується з 4 в ряд до 2 в ряд, а потім до 1 в стовпчик, зберігаючи рівномірні відступи. CSS Grid – ще потужніший інструмент побудови двовимірних сіток – дозволяє визначити

сітку з адаптивними треками. Зокрема, функція `repeat(auto-fit, minmax(200px, 1fr))` задає, що в контейнері може бути стільки стовпців шириною мінімум 200px, скільки поміститься, а решта простору ділиться рівномірно між ними. Це значить, що `grid` сам "накидає" потрібну кількість колонок залежно від ширини контейнера. У поєднанні з медіа-запитами (для грубого налаштування точок перелому) `Flexbox` і `Grid` дають розробникам неймовірну гнучкість без використання скриптів. Важливо, що ці технології теж частина стандартів W3C і добре підтримуються браузерами (`Flexbox` – з ~2013, `Grid` – з ~2017 років).

– Відносні одиниці та сучасні CSS-функції. Адаптивність потребує відмови від жорстко заданих піксельних розмірів. CSS3 ввів багато корисних одиниць: `%` – відсотки відносно батьківського елемента; `vw/vh` – відсотки від розмірів вікна браузера (`viewport width/height`); `rem` – відносно базового шрифту документа. Це дозволяє робити, наприклад, шрифти, що автоматично масштабуються під розмір екрану, або зображення з шириною `max-width: 100%` (щоб ніколи не виходили за межі батьківського блока). Також CSS3-функції на кшталт `calc()` дають можливість комбінувати відносні та абсолютні величини гнучко. Наприклад, `width: calc(50% - 20px)` – корисно для стовпців з відсотковою шириною і фіксованими відступами.

Зокрема на рисунку 3.1 подано приклад метатега `viewport` і медіа-запиту для `mobile-first` верстки.

```

1 <html>
2 <head>
3   <meta name="viewport"
4     content="width=device-wioth,
5     initial-scale=1.0">
6 </head>
7 <body>
8   /* Mobile-first styles
9   body {
10  }
11
12  @media (min-width: 768px)
13  {
14  }
15 </>

```

Рисунок 3.1 – Базові веб-технології

Загалом HTML5 та CSS3 надають все необхідне для побудови адаптивного веб-інтерфейсу. Більшість сучасних сайтів використовують підхід mobile-first: у CSS прописуються базові стилі, розраховані на маленькі екрани (мобільні), а потім за допомогою @media (min-width: ...) додаються стилі для більших екранів. Це логічно, оскільки легше почати з простої вертикальної компоновки для смартфона і ускладнювати/розширювати її для десктопу, ніж навпаки. Як зазначено в документації Bootstrap: “Bootstrap є mobile-first, ми використовуємо медіа-запити для створення сенсовних точок перелому для наших макетів та інтерфейсів” [17].

3.2 CSS-фреймворки

Розробка адаптивного сайту “з нуля” тільки засобами CSS може бути доволі трудомісткою. Щоб прискорити процес та впровадити кращі практики, веб-спільнота створила ряд CSS-фреймворків – готових бібліотек стилів і

компонентів, які забезпечують зі самого початку адаптивність макету. Найпопулярнішими наразі є Bootstrap та Tailwind CSS, хоча існують і інші (Foundation, Bulma, Materialize тощо). Розглянемо ці два як репрезентативні приклади різних підходів.

Bootstrap – один з найстаріших і найпоширеніших фронтенд-фреймворків (вперше випущений у 2011 році командою Twitter). Нині актуальна версія 5. Bootstrap пропонує:

- Готову 12-колонну сіткову систему з п'ятьма (у v5 – вже шістьма) типами брейкпойнтів: xs, sm, md, lg, xl (і xxl у v5). Макет будується через контейнери `<div class="container">`, рядки `<div class="row">` та стовпці `<div class="col">`. Ширина колонок задається класами з вказанням, скільки колонок із 12 займе блок на певному розмірі екрану. Наприклад, `<div class="col-xs-12 col-md-6">` означає, що елемент займатиме всю ширину (12 колонок) на малих екранах (xs, $<576\text{px}$) і половину ширини (6 колонок із 12) на середніх і вище (md, $\geq 768\text{px}$) [16]. Решта простору автоматично заповнюється іншими колонками або залишається порожньою. Така система реалізована через Flexbox і медіа-запити всередині bootstrap.css, тож розробнику не треба писати власні @media – достатньо скористатися відповідними класами. Bootstrap за замовчуванням орієнтований на mobile-first: стилі .col- застосовуються до всіх розмірів, а медіа-класи на зразок .col-md- перевизначають їх починаючи з даного брейкпойнту вгору.

- Набір готових адаптивних компонентів UI: навігаційні панелі (Navbar), випадаючі меню, модальні вікна, слайдери, таблиці, картки, сітки зображень тощо. Всі вони побудовані з урахуванням адаптивності. Наприклад, у Navbar меню автоматично ховається за "гамбургер"-іконкою на вузьких екранах, перетворюючись на випадаючий список – з мінімальним кодом з боку розробника. Таблиці можуть стати прокручуваними горизонтально у дрібних viewports, зберігаючи читабельність.

- Mobile-first підхід за замовчуванням. Bootstrap спочатку стилізує елементи під мобільний вигляд, а для великих екранів накладає додаткові стилі

через медіа-запити. Це видно у коді: базові класи (.container, .row, .col- тощо) розраховані на малий екран, а класи з префіксами md, lg додають зміни для ширших екранів [17]. Такий підхід узгоджується з рекомендаціями адаптивного дизайну і полегшує створення інтерфейсу, який “спочатку мобільний”.

– Препроцесорні змінні та міксіни. У Bootstrap легко налаштувати точки перелому, кількість колонок тощо, змінивши змінні SCSS. Це дає гнучкість, якщо потрібен нестандартний макет.

Bootstrap надає швидкий старт для адаптивної верстки. Недарма він широко використовується – навіть початківець, підключивши CDN-версію Bootstrap, може отримати результативний адаптивний сайт за лічені години. За допомогою компонентів Bootstrap багато типових UI-завдань (навігація, сітки карток, форм) вирішуються майже “з коробки” з уже закладеною адаптивністю.

Tailwind CSS – відносно новий (2017) CSS-фреймворк, який дотримується зовсім іншої філософії: utility-first. На відміну від Bootstrap, Tailwind не містить готових компонентів з фіксованим виглядом. Натомість він надає сотні утилітарних CSS-класів, кожен з яких відповідає за одну стилізуючу властивість (margin, padding, flex, color тощо). Розробник безпосередньо в HTML-розмітці комбінує ці класи для досягнення потрібного дизайну. Головне, що Tailwind чудово пристосований для адаптивності:

– Усі утиліті-класи можуть застосовуватись умовно для заданих брейкпойнтів за допомогою префіксів. За замовчуванням Tailwind має п’ять точок: sm ($\geq 640\text{px}$), md ($\geq 768\text{px}$), lg ($\geq 1024\text{px}$), xl ($\geq 1280\text{px}$), 2xl ($\geq 1536\text{px}$) [18]. Якщо перед класом додати префікс, він спрацює тільки при відповідній мінімальній ширині. Наприклад: `<div class="p-4 md:p-8 lg:p-12">` задасть внутрішні відступи 1rem (16px) на маленьких екранах, 2rem на $\geq 768\text{px}$ і 3rem на $\geq 1024\text{px}$. Tailwind сам згенерує потрібні медіа-запити в CSS. Інший приклад: клас `hidden md:block` – елемент буде прихований (`display: none`) на мобільних і показаний (`display: block`) на екранах md і більше. Таким чином, адаптивний дизайн у Tailwind досягається дуже легко і явно – просто додаванням потрібних брейкпойнт-префіксів до звичайних класів [18]. Документація фреймворку

підкреслює: “кожен утилітарний клас може бути застосований умовно на різних брейкпойнтах, що робить побудову складних адаптивних інтерфейсів надзвичайно простою” [18].

- Tailwind також mobile-first: стилі без префікса діють від найменших екранів, а префікси sm:, md: тощо перевизначають їх на більших.

- Значною перевагою є набори утиліт для layout: наприклад, класи flex, flex-col, flex-wrap, justify-center, items-baseline та безліч інших – все для управління flexbox; аналогічно для grid: grid, grid-cols-3 (3 колонки), gap-4 (відступи) і т.д. З їх допомогою можна створити адаптивну сітку, як у Bootstrap, але більш кастомізовану під конкретний дизайн. Tailwind не нав’язує 12-колонну систему – ви самі вирішуєте, скільки колонок і як вони поведуться.

- Tailwind включає утиліти і для типографіки, і для відступів, і для розмірів – все зручно масштабовано. Наприклад, класи text-lg md:text-2xl збільшать розмір шрифту на великих екранах; w-full md:w-1/2 зробить блок повної ширини на мобільці та 50% ширини на десктопі; space-y-4 md:space-y-0 md:space-x-4 – на мобільці елементи будуть вертикально з проміжком 1rem, а на десктопі перейдуть в рядок з горизонтальним проміжком. Tailwind дає дуже високу гранулярність контролю.

- Додатково, Tailwind має функцію Responsive variants – тобто можливість за замовчуванням генерувати певні варіанти класів для кожного брейкпойнту. Але зазвичай префікси достатні.

Таким чином, Tailwind підходить тим, хто хоче повної кастомізації дизайну, але все ж зручностей готової адаптивності. Він особливо популярний у сучасних SPA та React-проектах, де розробники хочуть швидко верстати прямо у компонентах без написання окремих CSS-файлів. Tailwind забезпечує, щоб інтерфейс одразу був адаптивним, і позбавляє від рутини написання медіа-запитів – це все беруть на себе утилітні класи [18].

Також на рисунку 3.2 наведено порівняння 12-колонної сітки Bootstrap та утилітарної сітки Tailwind CSS.



Рисунок 3.2 – CSS-фреймворки

Інші фреймворки. Bootstrap і Tailwind – не єдині. Foundation (Zurb) – колись головний конкурент Bootstrap – також надає адаптивну сітку та компоненти, але зараз менш популярний. Bulma – легший CSS-фреймворк без JS, теж mobile-first. Material-UI / MUI – це вже бібліотека компонентів для React, заснована на принципах Material Design від Google; вона також забезпечує responsive behavior компонентів (наприклад, Grid-система MUI для React використовує breakpoints аналогічно Bootstrap). В цілому, у 2020-х роках мало хто пише адаптивний інтерфейс з нуля – зазвичай або беруть CSS-фреймворк, або UI-бібліотеку під конкретний фреймворк (React, Vue). Це економить час і гарантує перевірену практиками адаптивність.

3.3 Інструменти дизайну та розробки

Окрім власне веб-технологій (HTML/CSS/JS), вагому роль у створенні якісного адаптивного дизайну відіграють інструменти для проектування інтерфейсів та сучасні фронтенд-фреймворки. Вони допомагають на етапах планування UX, макетування, а також при розробці динамічних веб-додатків.

Figma та інші UI/UX-дизайн інструменти. На етапі дизайну макетів дуже важливо продумати, як інтерфейс буде виглядати на різних розмірах екрану. Раніше дизайнери часто створювали окремо три макети: для мобілу (наприклад, ширина 360px), планшету (~768px) і десктопу (1200px). Сучасні інструменти дозволяють спростити цю задачу. Figma – один з найпопулярніших зараз сервісів для спільного дизайну інтерфейсів – має потужний механізм Auto Layout та Constraints, які фактично емулюють принципи адаптивності прямо у дизайн-макеті. Auto Layout в Figma дозволяє визначити контейнери, в яких елементи автоматично вирівнюються і змінюють розмір при зміні розміру контейнера. Це дуже схоже на flexbox/grid в CSS, але в середовищі дизайну. Завдяки цьому дизайнер може створити, приміром, панель навігації з Auto Layout, де кнопки рівномірно розподіляються. Якщо потім розширити або звужити фрейм, кнопки автоматично перебудуються (а при певному порозі можна вручну задати інший стиль – наприклад, сховати деякі кнопки за меню). Constraints в Figma – це налаштування прив'язок елементів до країв батьківського фрейму, що відповідає поведінці при масштабуванні. Наприклад, можна вказати, що елемент завжди тримається за лівий і правий край з певними відступами – тоді при розтягуванні екрану він буде розширюватися. По суті, Figma дозволяє прототипувати адаптивний дизайн: дизайнер робить один макет, а потім може перевіряти його в різних розмірах, і компоновка буде динамічно підлаштовуватись. Це значно економить час, бо не потрібно вручну перемальовувати три варіанти – достатньо налаштувати правила Auto Layout. Як зазначають експерти, Figma's Auto Layout “дає змогу дизайнерам створювати гнучкі та респонсивні дизайни, що адаптуються до змін контенту”. Він фактично працює як CSS Flexbox у візуальному редакторі [19]. Завдяки цьому можна легко побудувати, скажімо, компонент картки, що автоматично збільшується, якщо текст довший, і перевірити, як декілька таких карток переструктуровуються на ширшому фреймі (через Auto Layout + Wrap). Висновок: сучасний дизайн-процес тісно інтегрований з адаптивністю ще на

стадії прототипування, а Figma є провідним інструментом для цього завдяки своїм функціям адаптивного компоновання.

Відобразимо на рисунку 3.3 демонстрацію адаптивного компонента картки товару з Auto Layout та constraints.

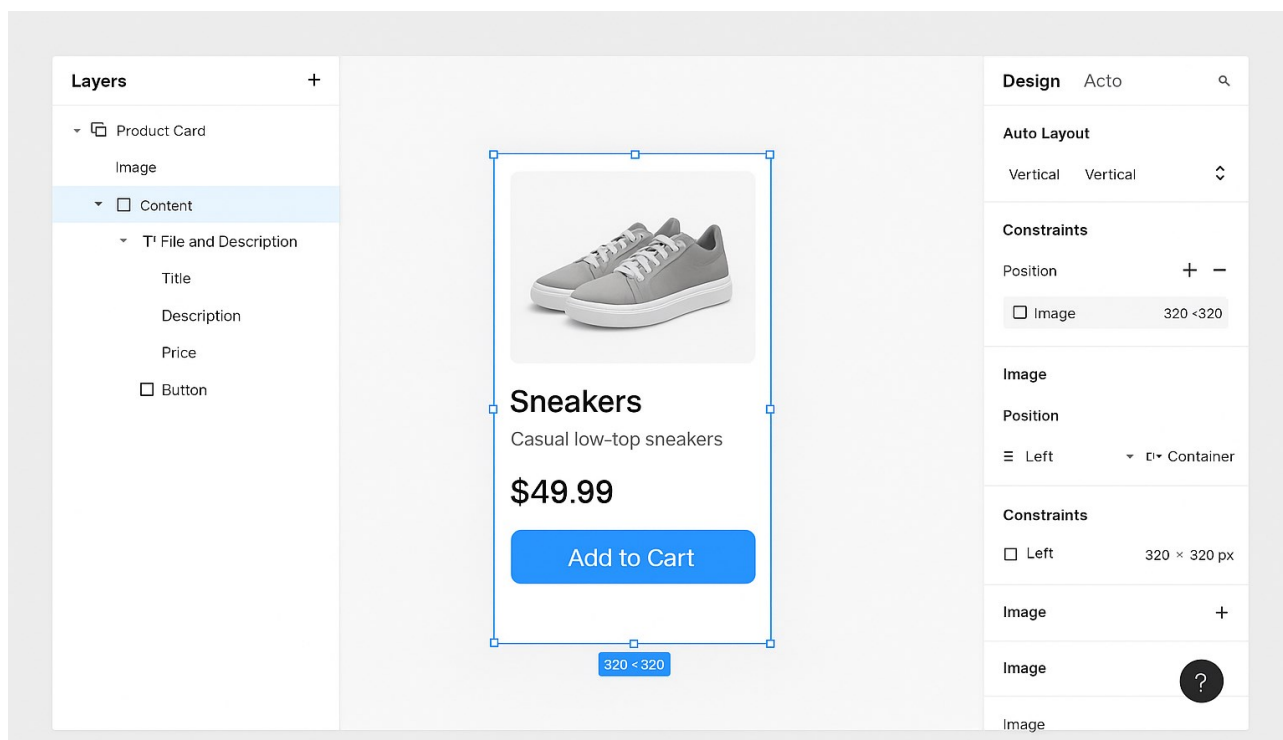


Рисунок 3.3 – Інструменти дизайну та розробки

Фронтенд-фреймворки (React, Angular, Vue). З точки зору програмної реалізації, зараз більшість веб-додатків будуються з використанням JS-фреймворків – найпопулярніший з них React. Сам по собі React – це бібліотека для побудови інтерфейсу на основі компонентів; він не визначає стилі і не займається адаптивністю напрямую. Однак екосистема React має багато готових рішень для адаптивного дизайну:

– По-перше, React легко інтегрується з CSS-фреймворками. Існує React-Bootstrap – набір готових компонентів React, стилізованих через Bootstrap. Вони несуть у собі всю адаптивну логіку. Як приклад, `<Container><Row><Col md={6} xs={12}>...</Col><Col md={6} xs={12}>...</Col></Row></Container>` – JSX-синтаксис, що відповідає бутстрапівській розмітці двох колонок, котрі на

смартфонах будуть на всю ширину, а на середніх екранах і більше ділитимуть екран навпіл [16]. React-компоненти Bootstrap під капотом мають ті ж класи або медіа-стилі. Таким чином, програміст на React може користуватися перевагами адаптивного Bootstrap через знайомий компонентний підхід [16].

– По-друге, є бібліотеки на кшталт `react-responsive`, що надають засоби умовного рендерингу компонентів на основі медіа-запитів. Наприклад, можна в React-кодi написати `<MediaQuery minWidth={768}>` (контент для десктопу) `</MediaQuery>` і `<MediaQuery maxWidth={767}>` (контент для мобілу) `</MediaQuery>`. Це дозволяє взагалі не писати CSS `@media`, а керувати адаптивністю логікою компонентів (хоча такий підхід інколи призводить до завантаження зайвого коду, його треба застосовувати обережно).

– CSS-in-JS та `Styled Components`. У React-проектах популярне стилювання компонентів через JS. Бібліотека `Styled-Components` дозволяє створювати стилі з врахуванням пропсів, у тому числі розмірів. Можна, наприклад, зробити стилі `const Box = styled.div\@media (max-width: 768px) { ... }`; – тобто включати медіа-запити безпосередньо в JS-файлі. Це синтаксичний цукор, але декому зручніше мати все в одному місці.

– `Material-UI (MUI)` для React, як згадувалось, включає `Grid`-компонент. Його API схоже: `<Grid container spacing={2}><Grid item xs={12} md={6}>...</Grid> ...</Grid>`. MUI під капотом генерує CSS з медіа-запитами, тож це альтернатива Bootstrap, але в стилі `Material Design` від Google. `Material Design`, до речі, сам визначає адаптивні правила – наприклад, рекомендує 4-колонну сітку на телефонах, 8-колонну на планшетах, 12-колонну на вебi. Компоненти MUI вже налаштовані відповідно до цих правил.

Інші технології. В контексті адаптивності можна також згадати:

– `Responsive Web Design Testing Tools` – наприклад, вбудований у `Chrome DevTools` режим емуляції різних пристроїв, сервіси на зразок `BrowserStack` (для тестування макетів на реальних девайсах) [16]. Вони важливі у розробці, щоб перевірити, як адаптивний дизайн працює на практиці.

– Продуктивність і адаптивність. Інструменти типу Lighthouse (Google) допомагають виявити проблеми, що можуть впливати на мобільний досвід: довгий час завантаження, непристосованість до повільних мереж тощо. Адаптивний дизайн має йти пліч-о-пліч з оптимізацією – наприклад, використанням `loading="lazy"` для зображень, щоб на мобільному не тягнути одразу весь контент.

– Frameworks/Generators. Такі системи як Bootstrap Studio або Webflow – інструменти, що дозволяють візуально верстати адаптивні сайти, генеруючи код. Вони використовують підкапотом ті ж CSS-grid та flex, просто надають зручний UI для створення адаптивного макету без кодування (drag-and-drop). Це теж частина екосистеми, але більш нішева (орієнтована на тих, хто не пише код).

3.4 Етапи реалізації адаптивного інтерфейсу

Крок 1: Аналіз та планування. Спершу проводиться аудит поточного інтерфейсу на різних пристроях. Використовуючи інструменти тестування (Chrome DevTools, BrowserStack тощо), команда виявила типові проблеми: на смартфонах сторінки магазину виглядали зменшеними, текст дрібний, елементи меню накладаються один на одного, картинки товарів вирізані по краях. Також з Google Analytics було помітно: мобільний трафік мав високий показник відмов (~70%) і короткий середній час сеансу (~30 секунд). Було прийнято рішення провести повний адаптивний редизайн з підходом mobile-first. Визначено ключові точки перелому дизайну: 576px (смартфони в портретній орієнтації), 768px (планшети/маленькі ноутбуки), 1024px (звичайні ноутбуки), 1200px і вище (великі монітори). Також поставлено цілі: знизити mobile bounce rate хоча б до 50%, підвищити конверсію з мобільних (додання товару в кошик) на 20%.

Крок 2: Проєктування UX і макетування. Дизайнери у Figma створили нові макети сторінок магазину. За допомогою функцій Auto Layout вони одразу

змоделювали поведінку блоків: наприклад, сітку товарів зробили 2-колонною для мобільного (щоб картки були достатньо великі), 3-колонною для планшета і 4-колонною для десктопу. Навігаційну панель спроектували з «гамбургер»-меню: при ширині $<576\text{px}$ вона ховає пункти меню і показує іконку, при натисканні на яку впливає повноекранне меню. В Figma вдалося інтерактивно перевірити, як довгі назви товарів переносяться на менших екранах, як виглядає карусель акцій на різних розмірах (вирішено на мобільному показувати 1 слайд, на десктопі – 3 одночасно). В макети заклали також адаптивні шрифти: наприклад, заголовки $h1$ – 24px на моб, 32px на десктоп. В результаті створено дизайн-систему з компонентами, що мають варіації для основних брейкпойнтів.

Крок 3: Верстка і розробка. Розробники обрали поєднання HTML5 + Tailwind CSS + React для реалізації. Спочатку зверстали адаптивний каркас сторінок:

- В HTML додали `<meta name="viewport" content="width=device-width, initial-scale=1">` у кожній сторінці, щоб мобільні пристрої відображали сторінку 1:1 [10].

- Використовуючи Tailwind CSS, швидко створили адаптивні сітки: контейнер з класом `max-w-6xl mx-auto` (центрування і обмеження ширини), сітка товарів з класами `grid grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-4` – це забезпечило 2 колонки на моб, 3 на $\geq 768\text{px}$, 4 на $\geq 1024\text{px}$ автоматично. Ніяких медіа-запитів вручну писати не довелося – Tailwind згенерував їх на основі цих утилітних класів [18].

- Навігаційну панель реалізували на React: створили компонент `Navbar`, що містив лого і меню. Меню – список посилань. Використали Tailwind-класи `hidden md:flex` для самого списку (при `md` і вище показується як горизонтальний `flex`), та окремий компонент `MobileMenu` з класом `md:hidden` (видимий лише на моб). При натисканні на іконку (кнопка з класом `md:hidden` для мобілки) викликається стан `openMobileMenu`, і компонент `MobileMenu` рендериться (включає вертикальний список пунктів з великими відступами для зручного тач-кліку).

– Головний банер (hero) зробили з адаптивним зображенням: застосували елемент `<img>` з атрибутами `srcset` – зображення 1200px ширини для десктопів, 800px для планшетів, 400px для мобільних, плюс атрибут `sizes="(min-width: 768px) 768px, 100vw"` – так браузер завантажить оптимальне за розміром зображення під поточний пристрій [14].

– Для типографіки використали Tailwind-класи, що масштабуються: основний текст `text-base md:text-lg`, заголовки `text-2xl md:text-4xl font-bold` тощо.

– Карусель акцій зробили з допомогою бібліотеки Swiper.js, яка підтримує параметр `slidesPerView` залежно від брейкпойнтів (напр., `{ 0: {slidesPerView: 1}, 768: {slidesPerView: 3} }`). Це забезпечило показ 1 слайду на малих екранах та 3 – на більших автоматично.

Крок 4: Тестування на різних пристроях. Після верстки провели ретельне тестування. За допомогою BrowserStack перевірили:

– На iPhone SE (320px ширина) – чи все читається, чи немає горизонтального скролу. Виявили пару проблем: деякі довгі слова виходили за межі карток – вирішили додати клас `break-words` в Tailwind, щоб переносилися. Кнопки трохи дрібнувати – збільшили `py-2 px-4` до `py-3 px-6` на мобільних.

– На iPad (768px) в портретній – побачили, що меню вже показується повністю (`md:flex`), але місця замало і останній пункт переноситься вниз. Вирішили: для `md` екранів зробити трохи менший шрифт у меню або скоротити текст пунктів. Внесли коригування.

– На десктопах 1920px – все виглядало добре, контент по центру, сітка товарів на 4 колонки з гарними відступами. Також протестували різні браузери (Chrome, Safari мобільний, Firefox) – завдяки використанню стандартизованих технологій (CSS Grid, flex, стандартні медіа) скрізь адаптивність працювала однаково.

Крок 5: Оптимізація продуктивності. Переконавшись у коректності UI, звернули увагу на швидкість: щоб мобільним користувачам не доводилося чекати, провели оптимізацію. Зображення стисли (JPEG progressive), включили

loading="lazy" для карток товарів. Tailwind CSS використовується у режимі JIT (just-in-time) – він включив в фінальний CSS тільки ті класи, що реально були використані, завдяки чому розмір стилів вийшов невеликим (~10KB gzipped). Також переконалися, що сторонні скрипти (наприклад, чат-підтримка) не псують UX на мобільному (при потребі вони відкладені до повного завантаження контенту).

Крок 6: Розгортання і збір аналітики. Оновлену версію веб-додатку було розгорнуто. Після цього протягом кількох тижнів моніторили показники Google Analytics і збирали відгуки користувачів.

Наведено послідовність етапів: аудит → прототипування → верстка → реліз на рисунку 3.4.

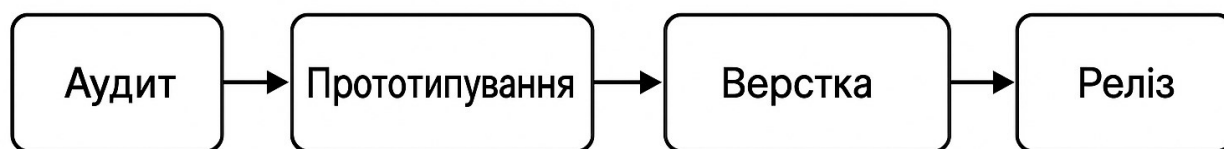


Рисунок 3.4 – Етапи практичної реалізації

3.5 Результати та аналіз ефективності

Рефакторинг веб-додатку під адаптивний дизайн приніс помітні поліпшення:

- Показник відмов мобільного трафіку за місяць після релізу знизився з ~70% до ~45%. Це колосальне падіння на 25 пунктів, що означає: суттєво більша частка мобільних відвідувачів почала взаємодіяти зі сторінками далі, ніж одна сторінка [14]. На десктопі bounce rate теж трохи покращився (~5% зниження), що може бути пов'язано з загальним покращенням UX (новий дизайн був більш легким і структурованим).

- Середня тривалість сесії мобільних користувачів зросла приблизно в 1,5 раза: з ~40 секунд до ~60 секунд. Кількість сторінок за сеанс також піднялася: середнє ~3 сторінки (було ~1,8). Це підтверджує тенденцію, що

адаптивний дизайн “затягує” користувача глибше у контент, роблячи перегляд комфортнішим [14].

- Конверсія (додавання товару у кошик) з мобільного трафіку збільшилась на ~30% відносно попереднього періоду. Якщо раніше лише 5% сесій мобільних доходили до додавання в кошик, то після редизайну – близько 6.5–7%. Це дуже значуще підвищення для e-commerce. В абсолютних числах, продажі з мобільних виросли. Цьому сприяли зручніші кнопки "Додати в кошик", які стали помітними і великими на екрані смартфона, та спрощений процес оформлення (односторінковий, оптимізований під мобільну клавіатуру).

- Поведінкові патерни за даними аналітики: теплові карти кліків показали, що на мобільній версії тепер користувачі активно взаємодіють з меню (прихованим за іконкою – багато кліків на “гамбургер”), гортануть карусель акцій (теж фіксувались взаємодії), переглядають список категорій (промотують). Раніше майже всі кліки були на перший екран і потім вихід. Це свідчить про покращене залучення.

- Відгуки користувачів (зібрані через випадкове опитування NPS та фідбек-форму) стали більш позитивними: багато хто відзначав, що “нарешті сайтом зручно користуватися з телефону”, “інтерфейс чистий і сучасний”. NPS серед мобільних користувачів зріс з 20 до 35, тобто стало більше тих, хто готовий рекомендувати сайт знайомим.

- Технічні метрики: Бал Google Lighthouse “Best Practices” піднявся до 100 (було ~80), “Accessibility” – теж близько 100, а “Performance” – ~90 на мобільній мережі (було 60). Це частково за рахунок оптимізації зображень, але й за рахунок усунення горизонтального скролу та дрібного тексту – Lighthouse раніше лаявся на ці аспекти. Показник “Mobile Friendly” тесту Google – пройдено (раніше сайт не проходив через елементи, що виходили за екран).

- SEO: Через деякий час покращились позиції в мобільному пошуку Google по деяких ключових словах. Це очікувано, адже Google віддає перевагу сайтам, оптимізованим під мобільні пристрої (mobile-first indexing). Таким чином, адаптивний дизайн навіть привів додатково нових користувачів.

На основі цих результатів можна зробити висновок, що практична реалізація адаптивного дизайну дала саме ті ефекти, які описувалися в теоретичній частині роботи:

- Користувачі отримали зручніший інтерфейс → зросли їхні ангажованість і задоволеність.
- Показники взаємодії (час, сторінки, CTR на елементи) покращились → вище залученість, більше шансів на конверсії.
- Бізнес-метрики (конверсії, продажі) на мобільних підтягнулись ближче до десктопних, зменшився розрив → веб-додаток став ефективнішим загалом.
- Реалізація технічно не була надто складною завдяки вибору сучасних інструментів (Tailwind, React) і бест-практисів. Час редизайну окупився підвищенням KPI.

На рисунку 3.5 наведено зростання конверсії та зниження bounce rate після впровадження адаптивного дизайну.

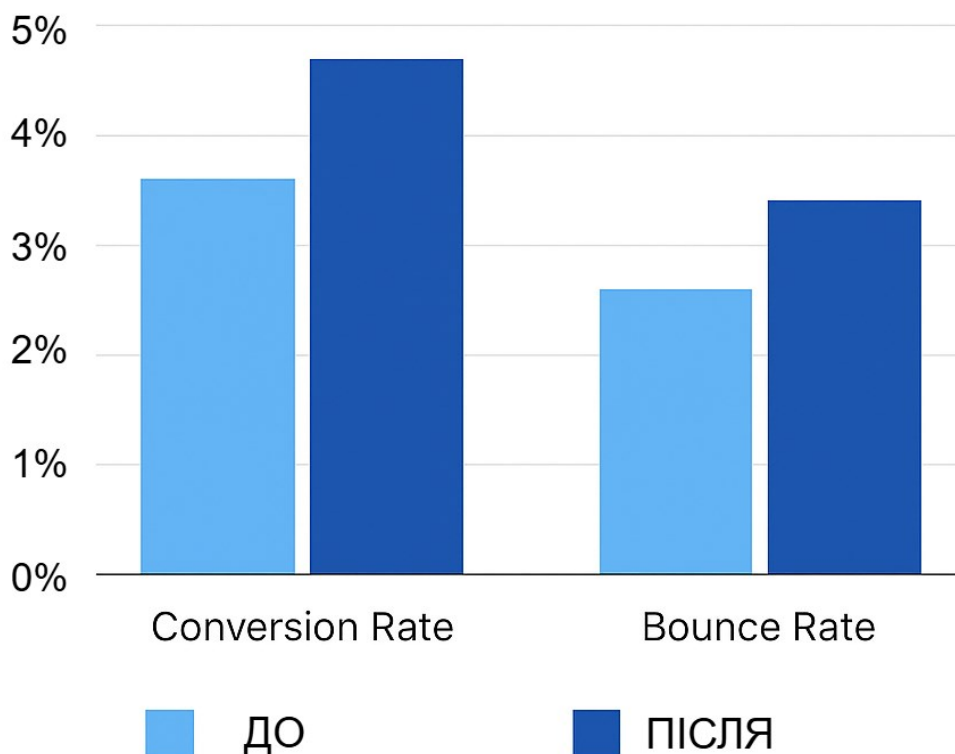


Рисунок 3.5 – Результати та аналіз ефективності

3.6 Висновки до третього розділу

Сьогодні розробники мають у своєму арсеналі потужні технології для створення адаптивних веб-додатків з найменшими зусиллями. Стандарти HTML5 і CSS3 забезпечують усю необхідну функціональність на рівні браузера – від мета-тегів viewport до медіа-запитів і гнучких макетів, які стали фундаментом responsive web design. На цьому фундаменті зведено безліч фреймворків (Bootstrap, Tailwind та ін.), що інкапсулюють типові рішення і прискорюють розробку, гарантуючи дотримання найкращих практик адаптивності. Інструменти для дизайну, як Figma, дозволяють продумувати responsive-стратегію інтерфейсу ще на етапі прототипування, завдяки функціям автоматичного компонування. А сучасні фронтенд-фреймворки типу React легко інтегруються з адаптивними стилями та компонентами, спрощуючи підтримку складних динамічних веб-додатків. Сукупно всі ці засоби роблять адаптивний дизайн не окремим “викликом”, а природною частиною процесу розробки: створити веб-додаток, який однаково успішно працює на смартфоні і на великому моніторі, сьогодні значно простіше, ніж десять років тому, завдяки зрілості технологій і наявності готових рішень.

Практичний кейс впровадження адаптивного дизайну демонструє, що навіть існуючий веб-додаток можна якісно трансформувати під потреби мобільних користувачів, отримавши швидкі й відчутні вигоди. Важливо дотримуватися системного підходу – починаючи з аналітики і дизайну, потім етапів розробки з використанням відповідних технологій, і завершуючи ретельним тестуванням. Наведені результати підтверджують основну гіпотезу: адаптивний дизайн суттєво підвищує ефективність взаємодії користувачів з веб-додатком. У фінальному розділі підсумуємо здобуті теоретичні та практичні знання, а також окреслимо рекомендації для подальших досліджень і розробок у цій сфері.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Питання щодо безпеки життєдіяльності

Тема кваліфікаційної роботи освітнього рівня «Бакалавр» присвячена темі аналізу впливу адаптивного дизайну на ефективність взаємодії користувачів із веб-додаткам.

Характерною особливістю науково-технічного прогресу на сучасному етапі є створення автоматизованих систем управління на основі впровадження електронно-обчислювальної техніки, наслідками чого є:

- збільшення кількості об'єктів і їх параметрів, якими необхідно управляти;
- розвиток системи дистанційного управління об'єктами, про динаміку яких судять на основі сприйнятих сигналів від засобів відображення інформації;
- ускладнення і збільшення швидкості перебігу виробничих процесів;
- зміни в умовах праці;
- зміни в структурі і характері трудової діяльності, поява професії оператора, основними функціями якого є управління, налагодження і контроль роботи різноманітних автоматизованих систем;
- створення різних типів систем «людина-машина», функціонування яких залежить від взаємодії технічних пристроїв і діяльності людини в єдиному контурі регулювання.

Під системою «людина—машина» розуміють систему, яка включає людину-оператора (групу операторів) і сукупність технічних засобів, за допомогою яких здійснюється трудова діяльність. За структурою машинного компонента системи «людина-машина» можуть бути різного рівня складності:

- інструментальні системи «людина—машина», в яких технічними пристроями є інструменти і прилади;

- прості системи «людина—машина», які включають стаціонарні і нестаціонарні технічні пристрої і людину, що використовує ці пристрої;

- складні системи «людина—машина», в яких людина управляє сукупністю технологічно взаємопов'язаних, але різних за функціональним призначенням апаратів, пристроїв і машин. Керування технологічним процесом забезпечується локальними системами автоматичного управління;

- система «людина—машина» у вигляді системотехнічних комплексів. Це складні системи, в яких людина взаємодіє не тільки з технічними пристроями, але й іншими людьми. Системотехнічні комплекси можна подати як ієрархію більш простих людино-машинних систем.

За типом взаємодії людини і машини виділяються системи неперервної та епізодичної взаємодії. Незалежно від рівня автоматизації системи «людина—машина» основною ланкою є людина, яка ставить мету, планує, направляє і контролює весь процес її функціонування. Особливості діяльності оператора такі:

- керування великою кількістю об'єктів і параметрів, що зумовлює значні навантаження на нервово-психічні функції;

- сприймання, переробка інформації і прийняття рішень є суттю діяльності;

- необхідність декодування інформації, отриманої в закодованому вигляді від приладів, і співвіднесення її зі станом реального процесу або об'єкта;

- висока точність дій і швидкість прийняття рішень та здійснення управлінських функцій;

- висока відповідальність за дії і прийняті рішення;

- висока готовність до екстрених дій;

- обмежена рухова активність, незначні м'язові навантаження;

- сенсорна монотонія або політонія;

- сенсорні, емоційні та інтелектуальні перенавантаження.

Ефективність роботи оператора, як і всієї системи в цілому, залежить від раціонального розподілу функцій між людиною і машиною, виходячи з врахування їх переваг і обмежень.

В одних випадках можливості людини набагато більші, ніж у машини, в інших, навпаки, машина має переваги над людиною. Так, людина здатна працювати в несподіваних ситуаціях, їй властива висока гнучкість і адаптивність до мінливих зовнішніх впливів, може працювати по багатьох програмах. Для машини неможливо запрограмувати всі випадковості; їй властиві мала гнучкість і висока ціна багатоваріантної роботи. Людина здатна створити повне уявлення про події, явища за неповної інформації. У машини така можливість обмежена. Людина має великі можливості для вибору способів дії, може швидко використовувати резерви, виправляти помилки. У машини ці можливості обмежені. В той же час у людини обмежена «пропускна здатність» щодо сприймання і переробки інформації. У машини — навпаки. У людини зменшується працездатність внаслідок розвитку втоми, розсіюється увага, з'являється різні емоційні стани і переживання. Машина має практично постійну роботоздатність. Людина порівняно повільно і неточно виконує обчислювальні операції, в той час як для машини характерні висока швидкість і точність обчислень. В зв'язку з цим необхідна автоматизація тих функцій, які можуть бути виконані без участі людини. За людиною залишаються функції щодо прийняття рішень, що і зумовлює більшу ефективність праці людини порівняно з роботою автоматичних пристроїв. Мислячий оператор посідає центральне місце у великих системах управління [42].

До взаємодії людини і технічних засобів ставляться підвищені вимоги, що вимагає пристосування техніки до людини (конструювання машин з врахуванням людських можливостей) і людини до машини (добір і підготовка спеціалістів). Зазначені особливості операторської праці дозволяють виділити її в специфічний вид професійної діяльності, основні етапи якої такі.

— Сприймання інформації щодо об'єктів керування та навколишнього середовища, яка важлива для розв'язання завдань, поставлених перед системою

«людина-машина». Для цього оператор повинен помітити сигнали, виділити з їх сукупності найбільш важливі та розшифрувати. Внаслідок цих дій у оператора формується попереднє уявлення про стан керованого об'єкта. Якість сприймання інформації залежить від типу і кількості індикаторів, організації інформаційного поля, характеристик інформації.

— Оцінка і переробка інформації. На цьому етапі порівнюються задані і реальні режими роботи системи, здійснюється аналіз та узагальнення інформації, виділяються критичні об'єкти і ситуації. На основі вже відомих критеріїв важливості і терміновості встановлюється черговість обробки інформації. На оцінку і переробку інформації впливають спосіб кодування, обсяг і динаміка змін в системі, а також відповідність обсягів інформації можливостям пам'яті і мислення оператора.

— Прийняття рішення про необхідні дії на основі проведеного аналізу та оцінки інформації, а також на основі інших відомостей про мету і умови роботи системи, можливі способи дії, наслідки правильних і неправильних рішень. Ефективність прийнятого рішення залежить від типу задачі, складності логічних умов, алгоритму та кількості можливих варіантів рішення.

— Реалізація прийнятого рішення шляхом виконання певних дій або видачі відповідних розпоряджень. На цьому етапі окремими діями є перекодування прийнятого рішення в машинний код, пошук потрібного органу керування і маніпуляції з ним тощо. Виконання рухів залежить від кількості органів керування, їх типу і способів розміщення.

На кожному етапі необхідний контроль власних дій, який може бути інструментальним або візуальним, що забезпечує надійність роботи оператора. Перші два етапи називають отриманням інформації, інші два — її реалізацією. Отримання інформації відбувається через сприймання оператором інформаційної моделі об'єкта керування, тобто різних носіїв інформації. Після декодування сприйнятих сигналів формується логічне знання про керований процес, яке називають концептуальною моделлю. Концептуальна модель дає можливість операторові поєднувати в єдине ціле окремі частини керованого

процесу і на основі прийнятого рішення здійснити ефективні керівні дії, тобто реалізувати отриману інформацію.

Незважаючи на загальні риси діяльності оператора, можна виділити окремі види операторської праці, для яких характерні свої особливості.

Оператор-технолог безпосередньо включений в технологічний процес. Працює переважно в режимі негайного обслуговування, тобто слідує, контролює і регулює технологічний процес з метою підтримання його в заданих межах. Переважаючими є керівні дії, виконання яких регламентується інструкціями. Останні містять повний набір ситуацій і рішень. До цього типу відносяться оператори технологічних ліній, оператори по прийняттю і переробці інформації.

Оператор-спостерігач (контролер, диспетчер). В його діяльності важливе значення мають інформаційні та концептуальні моделі, а також процеси прийняття рішень. Керівні дії дещо простіші. Може працювати в режимі відкладеного обслуговування. Сюди належать оператори радіолокаційних станцій, диспетчери на різних видах транспорту.

Оператор-дослідник в значно більшій мірі використовує понятійний апарат мислення і досвід, закладені в концептуальну модель. Органи керування мають менше значення, а інформаційні моделі, навпаки, велике. Сюди належать користувачі обчислювальних систем, дешифрувальники різних об'єктів [42].

Оператор-маніпулятор, в діяльності якого велике значення має сенсомоторна координація і моторні (рухові) навички. Механізми моторної діяльності основні, хоча використовується і апарат понятійного та образного мислення. Функціями оператора-маніпулятора є керування роботами, машинами, транспортними засобами і т. ін.;

Оператор-керівник. Головну роль в діяльності відіграють інтелектуальні процеси. Належать організатори, керівники різних рівнів, особи, які приймають відповідальні рішення в людино-машинних комплексах і володіють знаннями, досвідом, інтуїцією.

Поведінка оператора в екстремальних умовах є проявом і результатом психологічної готовності до діяльності. Розрізняють загальну і ситуативну (як стан) готовність. Загальна готовність являє собою раніше набуті знання, навички, вміння, мотиви щодо виконання певної діяльності. Тимчасовий стан готовності — це актуалізація всіх сил, психологічних можливостей для успішних дій в даний момент. Готовність до успішних дій в екстремальних ситуаціях залежить від особистісних властивостей оператора, рівня його підготовки і повноти інформації про стан керованого об'єкта. Емоційне напруження оператора після виконання особливо відповідальної роботи супроводжується психічним виснаженням (функціональною астеною) різного ступеня. Відмічаються слабкість процесів збудження (недостатня рухливість, пасивність, сповільнення мислення) або гальмування (помірно виражений рух, неглибокий аналіз і оцінка подій). Такий стан може тривати протягом 1...3 год (рідше добу), після чого з'являються головні болі, стомленість, апатія, неглибокий сон. Відмічаються погіршення пам'яті, сприймання. Тривалі і сильні емоційні напруження оператора негативно впливають на його діяльність, часто призводять до нервово-емоційних зривів. Основними напрямками зменшення емоційного напруження і підвищення надійності роботи оператора є науково обґрунтований професійний відбір і тренування, врахування і погодження особливостей конструкцій машин з можливостями людини при проектуванні людино-машинних систем; організація робочого місця.

4.2 Питання з основ охорони праці

Загальні вимоги безпеки до виробничого обладнання та технологічних процесів

Основними складовими безпеки праці на виробництві є:

- безпечне виробниче обладнання;
- безпечні технологічні процеси;
- організація безпечного виконання робіт.

ГОСТ 12.2.003–91.— основний нормативний документ з загальних вимог безпеки до виробничого обладнання за виключенням обладнання, яке є джерелом іонізуючих випромінювань.

Вимоги безпеки до виробничого обладнання конкретних груп, видів, моделей розробляються відповідно до вимог ГОСТ 12.2.003–91 з урахуванням призна-чення, виконання та умов його експлуатації.

- Безпека виробничого обладнання забезпечується:
- вибором принципів дії, джерел енергії, параметрів робочих процесів;
- мінімізацією енергії, що споживається чи накопичується;
- застосуванням вмонтованих в конструкцію засобів захисту та інформації про можливі небезпечні ситуації;
- застосуванням засобів автоматизації, дистанційного керування та контролю;
- дотримання ергономічних, обмеженням фізичних і нервово психологічних навантажень працівників.

Виробниче обладнання, при роботі як самостійно, так і в складі технологічних комплексів, повинно відповідати вимогам безпеки на протязі всього періоду його експлуатації.

Виробниче обладнання, робота якого супроводжується виділенням шкідливих речовин чи мікроорганізмів або пожежо– та вибухонебез-печних речовин, повинно включати вмонтовані пристрої для локалізації цих виділень. При відсутності таких пристроїв, в конструкції обладнання мають бути передбачені місця для підключення автономних пристроїв локалізації виділень. За необхідності згадані пристрої мають бути виконані з урахуванням чинних вимог щодо стану повітря робочої зони та захисту довкілля [42].

Якщо виробниче обладнання являється джерелом шуму, ультра та інфразвуку, вібрації, виробничих випромінювань (електромагнітних, лазерних тощо), то воно повинно бути виконано таким чином, щоб дія на працюючих перерахованих шкідливих виробничих факторів не перевищувала меж, встановлених відповідними чинними нормативами.

Однією із складових безпеки виробничого обладнання є конструкція робочого місця, його розміри, взаємне розміщення органів управління, засобів відображення інформації, допоміжного обладнання тощо. При розробці конструкції робочого місця слід дотримуватись вимог ГОСТ 12.2.032–78, ГОСТ12.2.033–84, ГОСТ12.2.049–80, ГОСТ12.2.061–81 та інших чинних нормативів. При цьому розміри робочого місця і його елементів мають забезпечувати виконання операцій в зручних робочих позах і не ускладнювати рухи працюючих. Конструкція крісла і підставки для ніг мають відповідати ергономічним вимогам.

Повне чи часткове припинення енергопостачання з наступним його відновленням, а також пошкодження мережі управління енергопостачанням не повинно призводити до виникнення небезпечних ситуацій.

Засоби захисту, що входять в конструкцію виробничого обладнання, повинні: забезпечувати можливість контролю їх функціонування; виконувати своє призначення безперервно в процесі роботи обладнання; діяти до повної нормалізації відповідного небезпечного чи шкідливого фактора, що спричинив спрацювання захисту; зберігати функціонування при виході із ладу інших засобів захисту. За необхідності включення засобів захисту до початку роботи виробничого обладнання, схемою управління повинні передбачатись відповідні блокування тощо.

ГОСТ 12.3.002–75.— чинний нормативний документ з загальних вимог безпеки до виробничих процесів.

Безпека виробничих процесів визначається, у першу чергу безпекою обладнання, яка забезпечується шляхом урахування вимог безпеки при складанні технічного завдання на його проектування, при розробці ескізного й робочого проекту, випуску та випробуваннях випробного зразка й передачі його у серійне виробництво згідно ГОСТ 15.001—73

Основними вимогами безпеки до технологічних процесів є: усунення безпосереднього контакту працюючих з вихідними матеріалами, заготовками, напівфабрикатами, готовою продукцією та відходами виробництва, що є

вірогідними чинниками небезпек; заміна технологічних процесів та операцій, що пов'язані з виникненням небезпечних та шкідливих виробничих факторів, процесами і операціями, за яких зазначені фактори відсутні або характеризуються меншою інтенсивністю; комплексна механізація та автоматизація виробництва, застосування дистанційного керування технологічними процесами і операціями при наявності небезпечних та шкідливих виробничих факторів; герметизація обладнання; застосування засобів колективного захисту працюючих; раціональна організація праці та відпочинку з метою профілактики монотонності й гіподинамії, а також обмеження важкості праці; своєчасне отримання інформації про виникнення небезпечних та шкідливих виробничих факторів на окремих технологічних операціях (системи отримання інформації про виникнення небезпечних та шкідливих виробничих факторів необхідно виконувати за принципом пристроїв автоматичної дії з виводом на системи попереджувальної сигналізації); впровадження систем конт-ролю та керування технологічним процесом, що забезпечують захист працюючих та аварійне відключення виробничого обладнання; своє-часне видалення і знешкодження відходів виробництва, що є джерелами небезпечних та шкідливих виробничих факторів, забезпечення пожежної й вибухової безпеки [42].

Розташування виробничого обладнання, вихідних матеріалів, заготовок, напівфабрикатів, готової продукції та відходів виробництва у виробничих приміщеннях і на робочих місцях не повинно являти собою небезпеку для персоналу. Відстані між одиницями обладнання, а також між обладнанням та стінами виробничих приміщень, будівель і споруд повинна відповідати вимогам діючих норм технологічного проектування, будівельним нормам та правилам.

До факторів, що визначають умови праці, відносяться також раціональні методи технології і організації виробництва. Зокрема, велику роль відіграє зміст праці, форми побудови трудових процесів, ступінь спеціалізації працюючих при виконанні виробничих процесів, вибір режимів праці та

відпочинку, дисципліна праці, психологічний клімат у колективі, організація санітарного й побутового забезпечення працюючих відповідно до

До осіб, які допущені до участі у виробничому процесі, ставляться вимоги щодо відповідності їх фізичних, психофізичних і, в окремих випадках, антропометричних даних характеру роботи. Перевірка стану здоров'я працюючих має проводитися як при допуску їх до роботи, так і періодично згідно з чинними нормативами. Періодичність контролю за станом їх здоров'я повинна визначатися залежно від небезпечних та шкідливих факторів виробничого процесу в порядку, встановленому Міністерством охорони здоров'я.

У автоматизованому виробництві необхідне також суворе виконання вимог безпеки під час ремонту й налагодження автоматичних машин та їх систем. Одним з перспективних напрямків комплексної автоматизації виробничих процесів є використання промислових роботів.

4.3 Висновок до четвертого розділу

Дослідження, проведене в межах кваліфікаційної роботи, присвячене темі аналізу впливу адаптивного дизайну на ефективність взаємодії користувачів із веб-додаткам, виявило важливість забезпечення електробезпеки в приміщеннях з електронно-обчислювальними машинами (ЕОМ). На сучасному етапі науково-технічного прогресу автоматизовані системи управління стають невід'ємною частиною виробничих процесів, що зумовлює необхідність врахування численних аспектів взаємодії людини і технічних засобів.

Особлива увага була приділена системам "людина-машина", які виступають ключовим елементом сучасного виробництва. Визначено різні типи таких систем, залежно від складності взаємодії між оператором та технічними засобами, та наведено характеристики діяльності операторів у різних умовах. Важливою складовою є раціональний розподіл функцій між людиною та машиною, що дозволяє забезпечити ефективність та надійність роботи.

Основними вимогами безпеки до виробничого обладнання та технологічних процесів є безпечність конструкції, мінімізація енергії, що споживається, впровадження засобів захисту та автоматизації, а також врахування ергономічних принципів. Безпека виробничих процесів забезпечується також за рахунок комплексної механізації, автоматизації, застосування дистанційного керування та впровадження систем контролю та керування.

Таким чином, забезпечення електробезпеки в умовах розробки інформаційного сайту для вивчення технологій веб-дизайну та веб-програмування є критично важливим. Підвищення безпеки та ефективності роботи операторів, а також раціональна організація праці сприятимуть подальшому розвитку автоматизованих систем управління та підвищенню продуктивності промислових процесів.

ВИСНОВКИ

Адаптивний веб-дизайн на сьогодні є невід'ємною умовою успіху будь-якого сучасного веб-додатку. Проведене дослідження – теоретичний аналіз, підкріплений практичною реалізацією – дозволяє зробити такі узагальнення:

1. Адаптивний дизайн як концепція сформувався на основі принципів гнучких макетів і медіа-запитів, закладених у стандарти HTML5/CSS3. Він забезпечує автоматичне підлаштування інтерфейсу під параметри пристрою, що є відповіддю на вибухове зростання мобільного інтернет-користування. Теоретичні засади адаптивності (responsive/adaptive design) стали загальноприйнятою практикою в індустрії, а відповідні технології вбудовані в саму тканину сучасного вебу [6],[9]. Таким чином, адаптивний дизайн – не модний тренд, а базовий стандарт веб-розробки.

2. Вплив на користувацьку ефективність і UX є надзвичайно істотним. Адаптивний інтерфейс підвищує юзабіліті – користувачі легше читають контент, навігація зрозуміліша, взаємодія інтуїтивна навіть на малих екранах. Емпіричні дані показують сильну кореляцію між наявністю адаптивного дизайну і високими оцінками зручності та задоволеності користувачів [23-28]. Крім того, усуваються бар'єри, що заважали мобільним відвідувачам – горизонтальний скрол, необхідність масштабування, занадто дрібні клікабельні елементи. Все це прямо впливає на метрики поведінки: знижується bounce rate (до 50% і більше покращення) [11], зростає час, проведений на сайті (на 50–80%) [14], збільшується глибина перегляду і ймовірність здійснення цільових дій (конверсії +30% і більше) [14]. Отже, адаптивний дизайн довів свою ефективність у покращенні взаємодії користувача як якісно (кращий UX), так і кількісно (поведінкові та бізнес-метрики).

3. Вплив на залученість та утримання аудиторії. Завдяки адаптивності веб-додатки не втрачають мобільних користувачів – навпаки, перетворюють їх на лояльних клієнтів. Поліпшення first impression і зручності навігації приводить до того, що користувач частіше повертається на сайт і проводить там

більше часу, формуючи звичку. Це підкріплюється зовнішніми дослідженнями: більшість користувачів воліють повторно заходити на мобільно-дружні сайти, і навпаки, швидко покидають неадаптивні ресурси, часто назавжди [11]. Внаслідок цього впровадження адаптивного дизайну сприяє зростанню постійної аудиторії, підвищує показники ретеншну та життя користувача в системі.

4. Інструментарій для реалізації адаптивності є зрілим і різноманітним. В роботі проаналізовано сучасні технології: від фундаментальних (медіа-запити, flexbox, grid) до високорівневих (CSS-фреймворки Bootstrap, Tailwind; UI-бібліотеки; React-компоненти). Доступність цих інструментів означає, що бар'єр для впровадження адаптивного дизайну низький – кожен розробник має у своєму розпорядженні готові рішення. Правильне їх використання дозволяє скоротити час розробки і мінімізувати ризик помилок. Наприклад, застосування Bootstrap гарантує перевірену адаптивність макету з мінімальними зусиллями, а Tailwind дає свободу гнучкої кастомізації, теж спрощуючи роботу з брейкпойнтами [18]. На етапі дизайну інструменти як Figma з Auto Layout дозволяють ще до коду врахувати адаптивність, що веде до кращого результату на виході [19].

5. Практична цінність адаптивного дизайну підтверджена кейсом впровадження. Приклад інтернет-магазину показав, що після адаптивного редизайну всі цільові показники відчутно зросли, а негативні моменти взаємодії зникли. Це принесло не тільки поліпшення UX, а й прямий бізнес-ефект у вигляді збільшення продажів та кращого SEO-рейтингу. Важливо, що досягнуто це було без необхідності створювати окремий мобільний додаток чи підтримувати дві різні версії сайту – отже, адаптивний підхід ще й економічно вигідний у довгостроковій підтримці (є “один веб-сайт для всіх”, який легше оновлювати і масштабувати) [23-28].

Підсумовуючи, адаптивний дизайн позитивно впливає на всі аспекти взаємодії користувача з веб-додатком: зручність, залученість, задоволеність, довіру, й врешті – на показники успішності самого продукту. З огляду на

тенденції зростання мобільного сегменту і появи нових видів пристроїв, адаптивність інтерфейсу ставатиме ще важливішою. Наступними кроками розвитку можуть бути адаптації для носимих пристроїв, інтеграція з голосовими інтерфейсами та контекстна адаптивність (підлаштування не тільки до екрану, а й до умов використання, поведінки користувача). Проте основна філософія лишається сталою: в центрі дизайну – користувач та його досвід, а адаптивний дизайн – це засіб зробити цей досвід найкращим на будь-якому пристрої. Веб-додатки, що впроваджують такі принципи, мають очевидну перевагу в конкурентній боротьбі за увагу сучасного користувача.

Наукове дослідження теми підтвердило гіпотези про значний вплив адаптивного дизайну на ефективність взаємодії користувачів. Подальші дослідження можуть зосередитись на кількісному моделюванні цього впливу в різних доменах (наприклад, освітні платформи, фінтех-сервіси), а також на вивченні поєднання адаптивного дизайну з іншими UX-практиками (персоналізація, адаптивні контент-рекомендації). Практикам веб-розробки ця робота надає узагальнений огляд методів і інструментів, які варто застосовувати, аби створювати конкурентоспроможні, зручні та ефективні веб-додатки. На основі проведеного аналізу можна впевнено рекомендувати: інвестувати в адаптивний дизайн – означає інвестувати в успіх і стійкість свого веб-продукту в умовах сучасного мультиекранного світу.

В першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

- Подано стислий огляд теоретичних засад адаптивного веб-дизайну.
- Розглянуто еволюцію стандартів HTML 5 / CSS 3 та їхню роль у формуванні адаптивних інтерфейсів.
- Висвітлено ключові принципи mobile-first, гнучких сіток і медіа-запитів.
- Проаналізовано переваги та обмеження підходів responsive й adaptive design.

В другому розділі кваліфікаційної роботи:

- Досліджено вплив адаптивного інтерфейсу на показники юзабіліті та залученості користувачів.

- Обґрунтовано вибір технологічного стеку (HTML 5, CSS Grid, Tailwind CSS, React) для реалізації адаптивності.

- Сформовано методику оцінювання ефективності дизайн-рішень за поведінковими метриками Google Analytics.

В третьому розділі кваліфікаційної роботи:

- Розроблено прототип адаптивного веб-додатка з каталогом товарів.
- Запропоновано UI-компоненти з авто-layout та гнучкою типографікою для різних брейкпойнтів.

- Спроектовано архітектуру фронтенду з використанням компонентного підходу React.

- Протестовано працездатність і зручність інтерфейсу на реальних мобільних та десктопних пристроях.

У розділі «Безпека життєдіяльності, основи охорони праці» висвітлено особливості безпечної роботи з комп'ютерним обладнанням.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 Cen, Z., & Zhao, Y. (2024). Enhancing User Engagement through Adaptive Interfaces: A Study on Real-time Personalization in Web Applications. *Journal of Economic Theory and Business Management*, 1(6), 1–15. DOI: 10.70393/6a6574626d.323332
- 2 Parlakkiliç, A. (2022). Evaluating the effects of responsive design on the usability of academic websites in the pandemic. *Education and Information Technologies*, 27(1), 1307–1322. DOI: 10.1007/s10639-021-10650-9
- 3 Hussain, A., & Mkpojiogu, E. (2015). The effect of responsive web design on the user experience with laptop and smartphone devices. *Jurnal Teknologi*, 77(4), 41–47
- 4 Priyadarshini, P. (2024). The Impact of User Interface Design on User Engagement. *International Journal of Engineering Research & Technology*, 13(03), 1–8
- 5 Marcotte, E. (2010). Responsive Web Design. A List Apart. Режим доступу: <https://alistapart.com>
- 6 W3C. (2012). Media Queries. W3C Recommendation 19 June 2012. World Wide Web Consortium
- 7 W3C. (2024). Media Queries Level 3. W3C Recommendation 21 May 2024. World Wide Web Consortium.
- 8 Gilbertson, S. (2012). It's Official, CSS Media Queries Are a Web Standard. *Wired*, June 20, 2012
- 9 Berners-Lee, T. (2014). HTML5 Recommendation Press Release. World Wide Web Consortium. (Цитовано у: Lardinois, F. (2014). “W3C Declares HTML5 Standard Complete.” *TechCrunch*, Oct 28, 2014)
- 10 Google Developers. (2019). Responsive Web Design Basics. *web.dev* by Google. Last updated 2019-02-12
- 11 Statista. (2023). Device share of internet usage worldwide. Режим доступу: <https://www.statista.com>

- 12 Roobykon Software. (2023). Top 10 Reasons Users Leave Your Site, and You Lose Profits. Roobykon Blog
- 13 Forbes Advisor. (2023). Top Website Statistics Today. Forbes Advisor Business. (Mobile-friendly stat)
- 14 Google Analytics Support. (2025). GA4 Engagement rate and bounce rate. Google Analytics Help webbestpractice.co.uk
- 15 Web Best Practice. (2025). Exploring the Impact of Responsive Web Design on User Engagement. WebBestPractice Blog webbestpractice.co.uk
- 16 BrowserStack. (2025). How to make React App Responsive using react-responsive?. BrowserStack Guide, May 20, 2025
- 17 Bootstrap Documentation. (2023). Bootstrap 5 Grid system. getbootstrap.com
- 18 Tailwind Labs. (2022). Responsive Design – Tailwind CSS v3 Documentation. Режим доступа: tailwindcss.com
- 19 DesignProject. (2023). Figma’s Auto Layout: The Ultimate Guide to Responsive Design.
- 20 Interaction Design Foundation. (n.d.). Adaptive vs. Responsive Design. Режим доступа: interaction-design.org
- 21 Google Developers. (n.d.). Material Design – Responsive layout grid. material.io (Material Design Guidelines)
- 22 Nielsen Norman Group. (2020). Using Grids in Interface Designs. NN/g Article. (Explains adaptive grids in UX design)
- 23 Snelling, J., & Fingal, D. (2020). Online Learning Guidelines during Pandemic.
- 24 Knight, K. (2011). One Web for All Devices. HTML5 Rocks (Google)
- 25 Baturay, M., & Birtane, M. (2013). Responsive Web Design for Education Websites. (Benefits of RWD)
- 26 Kim, H. (2013). Responsive Web Design: Implications for SEO and Usability. International Journal of Web Design, 5(1), 45–50.

- 27 Masri, M., & Luján-Mora, S. (2012). Accessibility Guidelines for Mobile Web. W3C WAI. (Accessibility and RWD)
- 28 Ethan Marcotte. (2014). Responsive design and user's devices. (Цитовано в Parlakkiliç 2021)
- 29 Rogatnev, A. (2015). Responsive Web Design without horizontal scroll.
- 30 Fielding, J. (2014). Beyond Mobile: Responsive Design for TV and Consoles. *ACM Interactions*, 21(5), 66–70
- 31 Stanko, A., Palka, O., Matiichuk, L., Martsenko, N., & Matsiuk, O. (2021, September). Smart City: A Review of Model Architecture and Technology. In 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT) (Vol. 2, pp. 273–277). IEEE.
- 32 Duda, O., Mykytyshyn, A., Mytnyk, M., & Stanko, A. Information technology sets formation and" TNTU Smart Campus" services network support. Proceedings of the 3rd International Workshop on ITTAP 2023, Ternopil, Ukraine, Opole, Poland, November 22–24, 2023.
- 33 Никитюк, В. В., Тененський, М. В., & Орловська, А. В. (2023). Аналіз використання EDA для вирішення проблем сучасних застоснків та систем. XI конф. „Інформаційні моделі, системи та технології“, 89-90.
- 34 Гузеляк, О., Шевчук, Ю., Береженко, Б. М., & Боднарчук, І. О. (2022). Програмна архітектура в розподілених командах гнучких проєктів. Матеріали Х науково-технічної конференції „Інформаційні моделі, системи та технології “ТНТУ імені Івана Пулюя, 110-112.
- 35 Готович, В. А., & Ралік, І. Р. (2022). Програмне забезпечення на основі клієнт-серверної архітектури для обліку реалізації товарів в торгівлі. Матеріали XI Міжнародної науково-практичної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 126-126.
- 36 Бідюк, О., & Марценко, С. (2025). Методи та засоби інформаційної безпеки іт інфраструктур. *Herald of khmelnytskyi national university. Technical sciences*, 347(1), 47-58.

37 Волович, В., Береженко, Б. М., & Боднарчук, І. О. (2022). Задача проєктування програмної архітектури в процесах забезпечення якості. Матеріали X науково-технічної конференції „Інформаційні моделі, системи та технології “ТНТУ імені Івана Пулюя, 104-106.

38 Семенюк, В. О., & Литвиненко, Я. В. (2023). Огляд методів захисту текстової інформації. Матеріали XI науково-технічної конференції „Інформаційні моделі, системи та технології “, 112-112.

39 Козак, С., Микитишин, А., & Станько, А. (2025). Оптимізація зберігання та обробки даних для іот-систем екологічного моніторингу. Measuring and computing devices in technological processes, (1), 323-330.

40 Sverstiuk, A., Matiichuk, L., Polyvana, U., Stanko, A., & Nykytyuk, V. (2024). Analytical analysis of approaches to assessing the quality of life in smart cities.

41 Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання / укл.: Стручок В. С. Тернопіль: ФОП Паляниця В. А., 2022. 156 с.

ДОДАТКИ

Фрагмент коду «Адаптивна картка товару» (HTML5 + CSS3 Grid)

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8" />
  <!-- Ключовий метатег для mobile-first -->
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Адаптивна картка товару</title>
  <style>
    /* Глобальні налаштування */
    :root { --gap: 1rem; }
    *      { box-sizing: border-box; margin: 0; padding: 0; }
    body   { font-family: Arial, sans-serif;
              display: grid; place-items: center; padding: 2rem; }
    /* 1) MOBILE-FIRST-CITKA */
    .card-grid {
      display: grid;
      grid-template-columns: 1fr;      /* одна колонка до 600 px */
      gap: var(--gap);
      width: 100%;
      max-width: 1280px;
    }
    /* 2) КАРТКА ТОВАРУ */
    .card {
      background: #fff;
      border: 1px solid #ddd;
      border-radius: 8px;
      overflow: hidden;
      display: flex;
      flex-direction: column;
    }
    .card img      { width: 100%; height: auto; }
    .card-content  { padding: 1rem; flex: 1; }
    .card h3       { font-size: 1.25rem; margin-bottom: .5rem; }
    .card p        { font-size: .9rem; line-height: 1.4; }
    .price         { color: #c0392b; font-weight: 700; margin: 1rem 0; }
```

```

.btn {
  display: block;
  text-align: center;
  padding: .75rem;
  background: #2ecc71;
  color: #fff;
  text-decoration: none;
}
/* 3) MEDIA QUERIES — РОЗШИРЮЄМО МАКЕТ */
@media (min-width: 600px) {          /* ≥ 600 px: 2 колонки */
  .card-grid { grid-template-columns: repeat(2, 1fr); }
}
@media (min-width: 900px) {          /* ≥ 900 px: 4 колонки */
  .card-grid { grid-template-columns: repeat(4, 1fr); }
}
</style>
</head>
<body>
  <section class="card-grid">
    <!-- Приклад однієї картки; дублюйте, щоб побачити сітку -->
    <article class="card">
      
      <div class="card-content">
        <h3>Сучасні кросівки</h3>
        <p>Легкі та зручні для міського стилю та тренувань.</p>
        <div class="price">€ 2 199</div>
      </div>
      <a href="#" class="btn">Додати у кошик</a>
    </article>
  </section>
</body>
</html>

```